kaspersky

Kaspersky Next XDR Expert

© 2026 AO Kaspersky Lab

Contents

Kaspersky Next XDR Expert
Quick links
What's new
About Kaspersky Next XDR Expert
- Hardware and software requirements
  - Common requirements and considerations
  - Kaspersky Deployment Toolkit requirements
  - Requirements for hosts with KUMA services
  - OSMP Console requirements
  - Network Agent requirements
  - Virtualization configuration
- Compatible applications and solutions
- Basic concepts
  - Administration Server
  - Hierarchy of Administration Servers
  - Web Server
  - Network Agent
  - Administration groups
  - Managed device
  - Unassigned device
  - Administrator's workstation
  - Management web plug-in
  - Policies
  - Policy profiles
  - Tasks
  - Task scope
  - How local application settings relate to policies
  - Distribution point
  - Connection gateway
  - Core
  - Storage
  - Collector
  - Correlator
  - Events
  - Alerts
  - Incidents
  - Resources
  - Services
  - KUMA agents
  - About severity

Architecture of Open Single Management Platform
- SIEM component architecture

OSMP Console interface
Changing the language of the OSMP Console interface
Changing the OSMP Console home page
Adding and removing bookmarks
Licensing
About the End User License Agreement
About the license key
About the activation code
About the key file
License limits
Kaspersky Next XDR Expert license features
Activating Kaspersky Next XDR Expert
Adding a license key for the SIEM component
Viewing information about license keys in use
Renewing licenses for Kaspersky applications
Removing a license key of the SIEM component
About data provision
Data provision in Open Single Management Platform
Data provision in Kaspersky Unified Monitoring and Analysis Platform
Quick start guide
Deployment and initial setup of Kaspersky Next XDR Expert
Verifying correctness of the Kaspersky Next XDR Expert configuration
Using the threat monitoring, detection and hunting features
Example of incident investigation with Kaspersky Next XDR Expert
Deployment of Kaspersky Next XDR Expert
Hardening Guide
Managing infrastructure of Kaspersky Next XDR Expert
Connection safety
Accounts and authentication
Managing protection of Kaspersky Next XDR Expert
Managing protection of client devices
Event transfer to third-party systems
Configuring protection for managed applications
Deployment schemes
Multi-node deployment scheme
Single-node deployment scheme
Ports used by Kaspersky Next XDR Expert
Preparation work and deployment
Multi-node deployment: Preparing the administrator and target hosts
Single-node deployment: Preparing the administrator and target hosts
KUMA installation requirements
Preparing the hosts for installation of the KUMA services
Installing a database management system
Configuring the PostgreSQL or Postgres Pro server for working with Open Single Management Platform
Preparing the KUMA inventory file for installing KUMA services outside the Kubernetes cluster
Multi-node deployment: Specifying the installation parameters
Single-node deployment: Specifying the installation parameters

Specifying the installation parameters by using the Configuration wizard
Installing Kaspersky Next XDR Expert
Configuring internet access for the target hosts
Synchronizing time on machines
Configuring deployment from Endpoint Agents via an intermediate proxy node
Installing KUMA services inside the Kubernetes cluster
Installing KUMA services outside the Kubernetes cluster
Deployment of multiple Kubernetes clusters and Kaspersky Next XDR Expert instances
Installing and upgrading Kaspersky Next XDR Expert in an air-gapped environment
Pre-check of infrastructure readiness for deployment
Signing in to Kaspersky Next XDR Expert
Integration with Endpoint Agents
Sandbox preparation and installation
Preparing IT infrastructure for the Sandbox installation
Preparing a virtual machine for installing Sandbox
Preparing an installation disk image
Installing Sandbox
Step 1. Viewing the End User License Agreement and Privacy Policy
Step 2. Selecting a disk for installing Sandbox
Step 3. Configuring the minimum password length for the Sandbox administrator password
Step 4. Creating the Sandbox administrator account
Step 5. Configuring the management network interface
Step 6. Assigning a hostname and adding DNS server addresses
Step 7. Configuring a static network route
Integration with Sandbox
Kaspersky Next XDR Expert maintenance
Kaspersky Next XDR Expert components
Upgrading Kaspersky Next XDR Expert from version 1.4 to 2.0
Updating Kaspersky Next XDR Expert components
Changing the Kaspersky Next XDR Expert parameters without reinstalling
Adding and deleting primary and worker nodes of the Kubernetes cluster
Versioning the configuration file
Uninstalling Kaspersky Next XDR Expert
Manual uninstalling of Kaspersky Next XDR Expert components
Reinstalling Kaspersky Next XDR Expert after a failed installation
Reinstalling Kaspersky Next XDR Expert components
Operation diagnostics of the Kaspersky Next XDR Expert components
Obtaining log files of Kaspersky Next XDR Expert components
Viewing OSMP metrics
Monitoring the state of Kaspersky Next XDR Expert components
Configuring email notifications about Kubernetes cluster health events
Managing email notification recipients for Kubernetes cluster health events
Adding SMTP server connection parameters to the configuration file without reinstalling Kaspersky Next XDR Expert
Viewing Longhorn metrics
Viewing EDR dashboards and metrics
Viewing the Node Exporter Full dashboard

Service monitoring
Viewing KUMA metrics
KUMA metric alert triggering conditions
Storing diagnostic information about Kaspersky Next XDR Expert components
Obtaining trace files
Logging the launches of custom actions
Backup copying and restoration of Kaspersky Next XDR Expert data
Configuring the storage for a Kaspersky Next XDR Expert backup copy
Configuring and managing the backup schedule
Creating a backup copy of Kaspersky Next XDR Expert data manually
Viewing the backups of Kaspersky Next XDR Expert data
Restoring Kaspersky Next XDR Expert data from a backup copy
Stopping the Kubernetes cluster nodes
Using certificates for public Kaspersky Next XDR Expert services
Calculation and changing of disk space for storing Administration Server data
Viewing and changing disk space for KUMA Core and the Metrics service
Rotation of Kubernetes cluster certificates
Rotation of Registry access credentials
Adding hosts for installing the additional KUMA services
Replacing a host that uses KUMA storage
Working with Kaspersky Next XDR Expert deployed on virtual machines
Demonstration deployment of Kaspersky Next XDR Expert
Preparing the administrator and target hosts
Specifying the installation parameters
Migration to Kaspersky Next XDR Expert
About migration from Kaspersky Security Center Windows
Exporting group objects from Kaspersky Security Center Windows
Importing the export file to Kaspersky Next XDR Expert
Switching managed devices to be under management of Kaspersky Next XDR Expert
About migration from KEDR 7.1
Transferred data
Hardware requirements for data migration
Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using a separate server for Kaspersky Next XDR Expert 2.0
Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using the same server for Kaspersky Next XDR Expert 2.0
Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using the same server for Kaspersky Next XDR Expert 2.0 in a distributed solution
Scenario: Migration from KATA/KEDR 7.1 to separate KATA 8.0 and Kaspersky Next XDR Expert 2.0 solutions
Connecting Endpoint Agents to Kaspersky Next XDR Expert and telemetry collection server
Connecting Endpoint Agent to the root tenant
Connecting Endpoint Agent to child tenants
Scenario: Telemetry migration using external storage
Telemetry migration for a single-node deployment
Telemetry migration for a multitenancy mode
Scenario: Telemetry migration without external storage
Exporting data from KEDR 7.1

Preparing to run the import
Importing data to Kaspersky Next XDR Expert 2.0
General limitations
Setting up and working with the SIEM component
KUMA services
Services tools
Getting service identifier
Stopping, starting, checking status of the service
Restarting the service
Deleting the service
Viewing the table of active queries
Partitions window
Searching for related events
Service resource sets
Creating a storage
ClickHouse cluster structure
ClickHouse cluster node settings
Cold storage of events
Removing cold storage disks
Detaching, archiving, and attaching partitions
Creating a set of resources for a storage
Creating a storage service in the KUMA Console
Installing a storage in the KUMA network infrastructure
Backing up, restoring, and importing partitions
Scheduling partition backup
Manually backing up partitions
Importing partitions from an external KUMA instance
Restoring partitions from backup
Creating a correlator
Starting the Correlator Installation Wizard
Step 1. General correlator settings
Step 2. Global variables
Step 3. Correlation
Step 4. Enrichment
Step 5. Response
Step 6. Routing
Step 7. Setup validation
Installing a correlator in a KUMA network infrastructure
Validating correlator installation
Creating the correlator correlator-ng
Configuring the correlator correlator-ng
Supported data types in correlator-ng
Correlation rule selectors and expressions
Attributes of the 'standard' correlation rule
Data functions in correlator-ng

- Creating an event router
  - Starting the event router Installation Wizard
    - Step 1. General settings of the event router
    - Step 2. Routing
    - Step 3. Setup validation
  - Installing the event router on the server
- Creating a collector
  - Starting the Collector Installation Wizard
    - Step 1. Connect event sources
    - Step 2. Transportation
    - Step 3. Event parsing
    - Step 4. Filtering events
    - Step 5. Event aggregation
    - Step 6. Event enrichment
    - Step 7. Routing
    - Step 8. Setup validation
  - Installing a collector in a KUMA network infrastructure
  - Validating collector installation
  - Ensuring uninterrupted collector operation
    - Event stream control using rsyslog
    - Event stream control using nginx
  - Predefined collectors
- Creating an agent
  - Creating a set of resources for an agent
  - Creating an agent service in the KUMA Console
  - Installing an agent in a KUMA network infrastructure
    - Installing a KUMA agent on Linux assets
    - Installing an Open Single Management Platform agent on Windows assets
  - Automatically created agents
  - Update agents
  - Transferring events from isolated network segments to KUMA
    - Diode agent configuration file
    - Description of secret fields
    - Installing Linux Agent in an isolated network segment
    - Installing Windows Agent in an isolated network segment
  - Transferring events from Windows machines to KUMA
AI services
- Analyzing alerts and incidents with KIRA
  - Configuring integration with KIRA
  - Analyzing using KIRA
  - Possible errors of the Analyze using KIRA task
- Creating search queries in natural language by using KIRA
  - Creating search queries in natural language and converting them into SQL queries by using KIRA
  - Possible errors while creating a query in natural language
- Configuring DLL Hijacking detection

AI asset score and status
Installing the AI asset score and status service
Reissuing the AI service certificate
Settings of the AI asset score and status service
Configuring event sources
Configuring receipt of Auditd events
Installing KUMA collector for receiving Auditd events
Configuring the event source server
Configuring receipt of KATA/EDR events
Configuring export of KATA/EDR events to KUMA
Creating KUMA collector for receiving KATA/EDR events
Installing KUMA collector for receiving KATA/EDR events
Configuring Open Single Management Platform for export of events to the KUMA SIEM-system
Configuring KUMA collector for collecting Open Single Management Platform events
Installing KUMA collector for collecting Open Single Management Platform events
Configuring receiving Open Single Management Platform event from MS SQL
Creating an account in the MS SQL database
Configuring the SQL Server Browser service
Creating a secret in KUMA
Configuring a connector
Configuring the KUMA Collector for receiving Kaspersky Security Center events from an MS SQL database
Installing the KUMA Collector for receiving Kaspersky Security Center events from the MS SQL database
Configuring receipt of events from Windows devices using KUMA Agent (WEC)
Configuring audit of events from Windows devices
Configuring an audit policy on a Windows device
Configuring an audit using a group policy
Configuring centralized receipt of events from Windows devices using the Windows Event Collector service
Configuring data transfer from the event source server
Configuring the Windows Event Collector service
Granting permissions to view Windows events
Granting permissions to log on as a service
Configuring the KUMA Collector for receiving events from Windows devices
Installing the KUMA Collector for receiving events from Windows devices
Configuring forwarding of events from Windows devices to KUMA using KUMA Agent (WEC)
Configuring receipt of events from Windows devices using KUMA Agent (WMI)
Configuring audit settings for managing KUMA
Configuring an audit using a local policy
Configuring an audit using a group policy
Configuring data transfer from the event source server
Granting permissions to view Windows events
Granting permissions to log on as a service
Configuring receipt of PostgreSQL events
Installing the pgAudit plugin
Configuring a Syslog server to send events
Configuring receipt of IVK Kolchuga-K events
Configuring export of IVK Kolchuga-K events to KUMA

Configuring receipt of CryptoPro NGate events
Configuring export of CryptoPro NGate events to KUMA
Configuring receipt of Ideco UTM events
Configuring export of Ideco UTM events to KUMA
Configuring receipt of KWTS events
Configuring export of KWTS events to KUMA
Configuring receipt of KLMS events
Configuring export of KLMS events to KUMA
Configuring receipt of KSMG events
Configuring export of KSMG events to KUMA
Configuring receipt of PT NAD events
Configuring export of PT NAD events to KUMA
Configuring receipt of events using the MariaDB Audit Plugin
Configuring the MariaDB Audit Plugin to send MySQL events
Configuring the MariaDB Audit Plugin to send MariaDB Events
Configuring a Syslog server to send events
Configuring receipt of Apache Cassandra events
Configuring Apache Cassandra event logging in KUMA
Configuring receipt of FreeIPA events
Configuring export of FreeIPA events to KUMA
Configuring receipt of VipNet TIAS events
Configuring export of ViPNet TIAS events to KUMA
Configuring receipt of Nextcloud events
Configuring audit of Nextcloud events
Configuring a Syslog server to send Nextcloud events
Configuring receipt of Snort events
Configuring logging of Snort events
Configuring receipt of Suricata events
Configuring logging of Suricata events.
Configuring receipt of FreeRADIUS events
Configuring audit of FreeRADIUS events
Configuring a Syslog server to send FreeRADIUS events
Configuring receipt of VMware vCenter events
Configuring the connection to VMware vCenter
Configuring receipt of zVirt events
Configuring export of zVirt events
Configuring receipt of Zeek IDS events
Conversion of the Zeek IDS event log format
Configuring DNS server event reception using the ETW connector
Configuration on the Windows side
Configuring Windows event reception using Kaspersky Endpoint Security for Windows
Configuring receipt of Codemaster Mirada events
Configuring audit of the Codemaster Mirada system
Configuring receipt of Postfix events
Configuring Postfix to send events
Configuring a KUMA collector for receiving and processing Postfix events

Configuring receipt of CommuniGate Pro events
Configuring CommuniGate Pro to send events
Configuring a KUMA collector for receiving and processing CommuniGate Pro events
Configuring receipt of Yandex Cloud events
Configuring audit of Yandex Cloud events
Configuring export of Yandex Cloud events
Configuring receipt of Microsoft 365 events
Configuring receipt of the Kontinent encryption system events
Preparing to receive Kontinent events
Configuring the kuma-kont utility
Starting and managing the kuma-kont service
Settings in the kuma-kont-config.yaml configuration file
Integration with other solutions
Integration with Kaspersky Security Center
Configuring the data refresh interval for Kaspersky Security Center assets
Manual import of Kaspersky Security Center assets
Importing events from the Kaspersky Security Center database
Kaspersky Endpoint Detection and Response integration
Importing Kaspersky Endpoint Detection and Response events using the kafka connector
Importing Kaspersky Endpoint Detection and Response events using the kata/edr connector
Configuring the display of a link to a Kaspersky Endpoint Detection and Response detection in the KUMA alert
Integration with Kaspersky CyberTrace
Integrating CyberTrace indicator search
Configuring CyberTrace to receive and process requests
Creating event Enrichment rules
Integrating CyberTrace interface
Integration with Kaspersky Threat Intelligence Portal
Initializing integration
Requesting information from Kaspersky Threat Intelligence Portal
Viewing information from Kaspersky Threat Intelligence Portal
Updating information from Kaspersky Threat Intelligence Portal
Connecting over LDAP
Enabling and disabling LDAP integration
Adding a tenant to the LDAP server integration list
Creating an LDAP server connection
Creating a copy of an LDAP server connection
Changing an LDAP server connection
Changing the data update frequency
Changing the data storage period
Starting account data update tasks
Deleting an LDAP server connection
Kaspersky Industrial CyberSecurity for Networks integration
Configuring integration in KICS for Networks
Configuring integration in KUMA
Enabling and disabling integration with KICS for Networks
Changing the data update frequency

Special considerations when importing asset information from KICS for Networks
Changing the status of a KICS for Networks asset
Integration with Neurodat SIEM IM
Kaspersky Automated Security Awareness Platform
Creating a token in KASAP and getting a link for API requests
Configuring integration in KUMA
Viewing information about the users from KASAP and changing learning groups
Sending notifications to Telegram
Creating and configuring a Telegram bot
Creating a script for sending notifications
Configuring notifications in KUMA
UserGate integration
Configuring integration in UserGate
Preparing a script for integration with UserGate
Configuring a response rule for integration with UserGate
Integration with Kaspersky Web Traffic Security
Configuring integration in KWTS
Preparing a script for integration with KWTS
Configuring a response rule for integration with KWTS
Integration with Kaspersky Secure Mail Gateway
Configuring integration in KSMG
Preparing a script for integration with KSMG
Importing asset information from RedCheck
Configuring receipt of Sendmail events
Configuring Sendmail logging
Configuring export of Sendmail events
Managing KUMA
Configuring the KUMA Core
Proxies
Connecting to an SMTP server
Working with Kaspersky Security Center tasks
Creating KUMA tasks in Kaspersky Security Center
Starting Kaspersky Security Center tasks manually
Starting Kaspersky Security Center tasks automatically
Checking the status of Kaspersky Security Center tasks
KUMA logs
KUMA notifications
Editing a KUMA user profile
Working with tenants
Working with geographic data
Geodata format
Converting geographic data from MaxMind to IP2Location
Importing and exporting geographic data
Default mapping of geographic data

KUMA resources
Operations with resources
Creating, renaming, moving, and deleting resource folders
Creating, duplicating, moving, editing, and deleting resources
Linking resources to services
Updating resources
Configuring a custom source using Kaspersky Update Utility
Exporting resources
Importing resources
Tag management
Destinations
Destination, internal type
nats-jetstream type
Tcp type
Udp type
Http type
Diode type
Kafka type
File type
Storage type
Correlator type
Destination, eventRouter type
Predefined destinations
Normalizers
Event parsing settings
Extended event schema
Adding extended event schema fields
Editing extended event schema fields
Importing and exporting extended event schema fields
Deleting extended event schema fields
Using extended event schema fields in normalizers
Enrichment in the normalizer
Conditions for forwarding data to an extra normalizer
Supported event sources
Aggregation rules
Enrichment rules
Correlation rules
Standard correlation rules
Simple correlation rules
Operational correlation rules
Correlation rules of the 'periodical' type
Creating a correlation rule in the correlator-ng correlator
Settings in the correlation-rules.yaml file
Variables in correlators
Local variables in identical and unique fields
Local variables in selector
Local Variables in event enrichment

- Local variables in active list enrichment
- Properties of variables
- Requirements for variables
- Functions of variables
- Declaring variables
- Predefined correlation rules
- MITRE ATT&CK matrix coverage
- Filters
- Active lists
- Viewing the table of active lists
- Adding active list
- Viewing the settings of an active list
- Changing the settings of an active list
- Duplicating the settings of an active list
- Deleting an active list
- Viewing records in the active list
- Searching for records in the active list
- Adding a record to an active list
- Duplicating records in the active list
- Changing a record in the active list
- Deleting records from the active list
- Import data to an active list
- Exporting data from the active list
- Predefined active lists
- Dictionaries
- Response rules
- Response rules for Kaspersky Security Center
- Response rules for a custom script
- Response rules for KICS for Networks
- Response rules for Kaspersky Endpoint Detection and Response
- Active Directory response rules
- Virtual Administration Server
- Connectors
- Viewing connector settings
- Adding a connector
- Connector settings
- Tcp type
- udp type
- Netflow type
- Sflow type
- nats-jetstream type
- Connector, kafka type
- kata/edr type
- Connector, http type
- Sql type
- File type
- Type 1c-xml

Type 1c-log
Diode type
Ftp type
Nfs type
vmware type
Wmi type
Wec type
snmp type
snmp-trap type
Configuring the source of SNMP trap messages for Windows
Configuring and starting the SNMP and SNMP trap services
Configuring the Event to Trap Translator service
elastic type
etw type
universalAPI type
Predefined connectors
Secrets
Context tables
Viewing the list of context tables
Adding a context table
Viewing context table settings
Editing context table settings
Duplicating context table settings
Deleting a context table
Viewing context table records
Searching context table records
Adding a context table record
Editing a context table record
Deleting a context table record
Importing data into a context table
Exporting data from a context table
Building a hierarchy of OSMP instances
Connecting a child instance to the parent instance
Disconnecting a child instance from a parent instance
Integration with other solutions
Integration with Kaspersky Automated Security Awareness Platform
Creating a token in KASAP and getting a URL for API requests
Integration with Kaspersky Threat Intelligence Portal
Integration with KATA/KEDR
Configuring integration between Kaspersky Next XDR Expert and script launch service
Threat detection
Working with alerts
About alerts
Alert data model
Viewing the alert table
Viewing alert details
Assigning alerts to analysts

- Changing an alert status
- Creating alerts manually
- Linking alerts to incidents
- Unlinking alerts from incidents
- Linking events to alerts
- Unlinking events from alerts
- Working with alerts on the investigation graph
- Adding an IOA rule to exclusions
- Managing aggregation rules
  - Creating aggregation rules
  - Editing aggregation rules
  - Copying aggregation rules to another tenant
  - Deleting aggregation rules

Working with incidents

- About incidents
- Incident data model
- Creating incidents
- Creating child incidents
- Viewing the incident table
- Viewing incident details
- Assigning incidents to analysts
- Changing an incident status
- Changing an incident priority
- Merging incidents
- Exporting incidents to a file
- Creating Managed Detection and Response incidents from XDR incidents
- Investigation graph
  - Process trees
    - Building process trees
    - Deleting process trees
    - Commenting on process trees
- Segmentation rules
- Copying segmentation rules to another tenant
- Managing Service Level Agreement (SLA)
  - Creating Service Level Agreement (SLA) entries
  - Deleting Service Level Agreement (SLA) entries
- Managing incident types
  - Viewing the incident types table
  - Creating incident types
  - Editing incident types
  - Deleting incident types
- Managing incident workflows
  - Viewing incident workflows table
  - Predefined incident workflows
  - Creating incident workflows
  - Editing incident workflows and statuses
  - Deleting incident workflows

- Managing incident effort
  - Adding effort for incidents
  - Editing effort for incidents
  - Deleting effort for incidents
  - Generating and exporting incident reports
- Configuring the retention period of alerts and incidents
- Configuring the automatic assignment of alerts and incidents
- Viewing asset details
- Investigating and responding via remote terminal
  - Managing remote terminal connections
  - Performing response actions via remote terminal
- Threat hunting
  - Working with events
    - Event data model
      - Process
      - File Create
      - File Modify
      - File Rename
      - File Read
      - File Delete
      - File Attributes Modify
      - File Creation Time Modify
      - File Hardlink Created
      - File Symlink Created
      - Module
      - Registry Create
      - Registry Delete
      - Registry Modify
      - Registry Rename
      - Registry Query
      - Registry Save
      - Connection
      - Port Listen
      - AMSI Scan
      - Process Console Input
      - Driver
      - Process Interpreted File Run
      - DNS
      - Pipe Create
      - Pipe Connect
      - LDAP
      - WMI Activity
      - WMI Consumer Register
      - Code Injection
      - Process Access
      - Threat Detect
      - Threat Detect Processing Result

- Linux Event Log
- Windows Event Log
- Blocked Document
- Applock
- Process Terminated
- Device
- File System Mount
- File System Unmount
- Bits Job Create
- Bits Job Complete
- Bits Job Start
- Bits Job Stop
- Bits Job Add File
- Bits Job Error
- Service Create
- Service Modify
- Service Delete
- Scheduled Task Create
- Scheduled Task Modify
- Scheduled Task Delete
- File Owner Change
- File ACL Change
- Registry Owner Change
- Registry ACL Change
- Executable Memory Mapped
- Agent Start
- Agent Stop
- Agent Install
- Agent Uninstall
- Agent Action
- System Startup
- System Shutdown
- Connection Close

Granular access to events

Viewing the events table

Searching and filtering events

Manually creating SQL queries

Generating an SQL query using a builder

Viewing event details

Saving and selecting events filter configuration

Filtering events by time range

Exporting events

Retrospective scan

Getting events table statistics

Threat response
Response actions
Terminating processes
Moving devices to another administration group
Updating databases
Changing authorization status of devices
Moving files to quarantine
Restoring files from quarantine
Getting a file from quarantine
Deleting files from quarantine
Viewing information about KASAP users and changing learning groups
Enabling and disabling network isolation
Prevention rules
Performing the Add prevention rule and Delete prevention rule response actions
Working with the table of prevention rules
Creating a prevention rule
Importing a prevention rule
Running a program
Managing a service
Blocking IP addresses through Kaspersky NGFW
Getting files for further analysis
Getting a file
Getting a registry key
Getting NTFS metafiles
Getting forensics
Getting a disk image
Getting a memory dump
Getting a process memory dump
Scanning devices for threats
Running a malware scan
YARA scan
IOC scan
Scanning files obtained from devices
Deleting a file
Responding through Active Directory
Responding by using third-party scripts
Responding through UserGate
Responding through Ideco NGFW
Responding through Ideco UTM
Responding through Redmine
Responding through Check Point NGFW
Responding through Sophos Firewall
Responding through Continent 4
Responding through SKDPU NT
Responding through FortiGate
Responding through Palo Alto NGFW

- Viewing response history from alert or incident details
- Viewing the result of the malware scan
- Working with files obtained as a result of response actions
- Playbooks
  - Viewing the playbooks table
  - Creating playbooks
  - Configuring playbook execution steps in visual editor
  - Editing playbooks
  - Customizing playbooks
  - Viewing playbook properties
  - Terminating playbooks
  - Deleting playbooks
  - Launching playbooks in test mode
  - Launching playbooks and response actions
    - Launching playbooks manually
    - Launching playbooks for objects specified by users
    - Launching playbooks in the Training operation mode
  - Configuring manual approval of response actions
  - Approving playbooks or response actions
  - Enrichment from playbook
  - Viewing response history
  - Predefined playbooks
    - [KL] P001 "Creation of executable files by office applications"
    - [KL] P002 "Windows Event Log was cleared"
    - [KL] P003 "Suspicious child process from wmiprvse.exe"
    - [KL] P004 "Sandbox Medium Detect"
    - [KL] P005 "Sandbox High Detect"
  - Playbook trigger
  - Playbook algorithm
    - Playbook parameters
    - Execution step parameters
      - Loop
      - Parallel
      - Decision
      - UpdateData
    - ResponseFunction parameters
  - Editing incidents by using playbooks
  - Editing alerts by using playbooks
- Managing Sandbox
  - Initial Sandbox setup
    - Configuring database updates for Sandbox
    - Sandbox network interfaces
      - Configuring DNS settings
      - Configuring a management network interface
      - Configuring Internet access for processed objects
      - Adding a static network route
    - Changing the Sandbox administrator account password

Sandbox authentication policies
Terminating inactive Sandbox sessions
Notifications about failed Sandbox authentication attempts
Setting up sending detected objects to the KPSN denylist
Making virtual machines ready for work
Uploading images to Storage
Importing a template
Creating a custom virtual machine template
Configuring the operating system and software
Creating a virtual machine
Installing a virtual machine
Downloading debug symbols
Configuring OSMP to use detection technologies
Updating anti-virus databases for Sandbox and Anti-Malware Engine
Creating a list of passwords for password-protected archives
Selecting operating systems for scanning objects in Sandbox
Creating custom Sandbox rules
Configuring sending of files from Endpoint Agents to Sandbox
About indicators of compromise, IOC files and IOC rules
Importing IOC rules
Importing YARA rules
REST API
Creating a token
Authorizing API requests
Working with Open Single Management Platform
Configuring Administration Server
Configuring the Administration Server connection address
Configuring the connection of OSMP Console to Administration Server
Configuring internet access settings
Certificates for work with Open Single Management Platform
About Open Single Management Platform certificates
Requirements for custom certificates used in Open Single Management Platform
Reissuing the certificate for OSMP Console
Replacing certificate for OSMP Console
Converting a PFX certificate to the PEM format
Scenario: Specifying the custom Administration Server certificate
Replacing the Administration Server certificate by using the klsetsrvcert utility
Connecting Network Agents to Administration Server by using the klmover utility
Hierarchy of Administration Servers
Creating a hierarchy of Administration Servers: adding a secondary Administration Server
Viewing the list of secondary Administration Servers
Managing virtual Administration Servers
Creating a virtual Administration Server
Enabling and disabling a virtual Administration Server
Assigning an administrator for a virtual Administration Server
Changing the Administration Server for client devices
Deleting a virtual Administration Server

Configuring Administration Server connection events logging
Setting the maximum number of events in the event repository
Changing DBMS credentials
Backup copying and restoration of the Administration Server data
Configuring the Administration Server Backup task
Using the KDT utility to recover Administration Server data
Deleting a hierarchy of Administration Servers
Access to public DNS servers
Configuring the interface
Encrypt communication with TLS
Backup copying and restoration of KUMA Core under Kaspersky Next XDR Expert
Discovering networked devices
Scenario: Discovering networked devices
Windows network polling
IP range polling
Domain controller polling
Configuring a Samba domain controller
Using VDI dynamic mode on client devices
Enabling VDI dynamic mode in the properties of an installation package for Network Agent
Moving devices from VDI to an administration group
Managing client devices
Settings of a managed device
Creating administration groups
Device moving rules
Creating device moving rules
Copying device moving rules
Conditions for a device moving rule
Adding devices to an administration group manually
Moving devices or clusters to an administration group manually
About clusters and server arrays
Properties of a cluster or server array
Adjustment of distribution points and connection gateways
Standard configuration of distribution points: Single office
Standard configuration of distribution points: Multiple small remote offices
Calculating the number and configuration of distribution points
Assigning distribution points automatically
Assigning distribution points manually
Modifying the list of distribution points for an administration group
Enabling a push server
About device statuses
Configuring the switching of device statuses
Device selections
Viewing the device list from a device selection
Creating a device selection
Configuring a device selection
Exporting the device list from a device selection
Removing devices from administration groups in a selection

Device tags
Device tags
Creating a device tag
Renaming a device tag
Deleting a device tag
Viewing devices to which a tag is assigned
Viewing tags assigned to a device
Tagging a device manually
Removing an assigned tag from a device
Viewing rules for tagging devices automatically
Editing a rule for tagging devices automatically
Creating a rule for tagging devices automatically
Running rules for auto-tagging devices
Deleting a rule for tagging devices automatically
Data encryption and protection
Viewing the list of encrypted drives
Viewing the list of encryption events
Creating and viewing encryption reports
Granting access to an encrypted drive in offline mode
Transmitting encryption keys between Administration Servers
Changing the Administration Server for client devices
Viewing and configuring the actions when devices show inactivity
Remote access to managed devices
Remote access from a Linux-based device with OSMP Console to a Linux-based managed device
Remote access from a Linux-based device with OSMP Console to a Windows-based managed device
Remote access from a Windows-based device with OSMP Console to a Linux-based managed device
Remote access from a Windows-based device with OSMP Console to a Windows-based managed device
Managing assets
Adding an asset category
Configuring the table of assets
Searching assets
Exporting asset data
Viewing asset details
Adding assets
Adding asset information in OSMP
Importing asset information from Kaspersky Security Center
Importing asset information from MaxPatrol
Importing asset information from KICS for Networks
Examples of asset field comparison during import
Settings of the kuma-ptvm-config.yaml configuration file
Assigning a category to an asset
Editing the parameters of assets
Archiving assets
Deleting assets
Bulk deletion of assets
Updating third-party applications and fixing vulnerabilities on Kaspersky Security Center assets
Moving assets to a selected administration group

Asset audit
Configuring an asset audit
Storing and searching asset audit events
Enabling and disabling an asset audit
Custom asset fields
Critical information infrastructure assets
Deploying Kaspersky applications
Scenario: Kaspersky applications deployment
Protection deployment wizard
Step 1. Starting Protection deployment wizard
Step 2. Selecting the installation package
Step 3. Selecting a method for distribution of key file or activation code
Step 4. Selecting Network Agent version
Step 5. Selecting devices
Step 6. Specifying the remote installation task settings
Step 7. Restart management
Step 8. Removing incompatible applications before installation
Step 9. Moving devices to Managed devices
Step 10. Selecting accounts to access devices
Step 11. Starting installation
Adding management plug-ins for Kaspersky applications
Removing management web plug-ins
Viewing the list of components integrated in Open Single Management Platform
Viewing names, parameters, and custom actions of Kaspersky Next XDR Expert components
Downloading and creating installation packages for Kaspersky applications
Creating installation packages from a file
Creating stand-alone installation packages
Changing the limit on the size of custom installation package data
Installing Network Agent for Linux in silent mode (with an answer file)
Preparing a device running Astra Linux in the closed software environment mode for installation of Network Agent
Viewing the list of stand-alone installation packages
Distributing installation packages to secondary Administration Servers
Preparing a Linux device for installation of Network Agent
Installing applications using a remote installation task
Installing an application remotely
Installing applications on secondary Administration Servers
Specifying settings for remote installation on Unix devices
Starting and stopping Kaspersky applications
Replacing third-party security applications
Removing applications or software updates remotely
Preparing a device running SUSE Linux Enterprise Server 15 for installation of Network Agent
Preparing a Windows device for remote installation
Configuring Kaspersky applications
Scenario: Configuring network protection
About device-centric and user-centric security management approaches
Policy setup and propagation: Device-centric approach
Policy setup and propagation: User-centric approach

About switching Network Agent to other Administration Servers
Creating a connection profile for out-of-office users
Policies and policy profiles
About policies and policy profiles
About lock and locked settings
Inheritance of policies and policy profiles
Hierarchy of policies
Policy profiles in a hierarchy of policies
How settings are implemented on a managed device
Managing policies
Viewing the list of policies
Creating a policy
General policy settings
Modifying a policy
Enabling and disabling a policy inheritance option
Copying a policy
Moving a policy
Exporting a policy
Importing a policy
Forced synchronization
Viewing the policy distribution status chart
Activating a policy automatically at the Virus outbreak event
Deleting a policy
Managing policy profiles
Viewing the profiles of a policy
Changing a policy profile priority
Creating a policy profile
Copying a policy profile
Creating a policy profile activation rule
Deleting a policy profile
Exporting and importing a policy profile
Network Agent policy settings
Usage of Network Agent for Windows, Linux, and macOS: Comparison
Comparison of Network Agent settings by operating systems
Manual setup of the Kaspersky Endpoint Security policy
Configuring Kaspersky Security Network
Checking the list of the networks protected by Firewall
Disabling the scan of network drives
Excluding software details from the Administration Server memory
Configuring access to the Kaspersky Endpoint Security for Windows interface on workstations
Configuring the registration of policy events in the Administration Server database
Manual setup of the group update task for Kaspersky Endpoint Security
Kaspersky Security Network (KSN)
About KSN
Setting up access to KSN
Enabling and disabling the usage of KSN
Configuring the connection to a local reputation database of KPSN

Viewing the accepted KSN Statement
Accepting an updated KSN Statement
Checking whether the distribution point works as KSN proxy server
Viewing the statistics of KSN proxy server
Managing endpoint protection tasks
About tasks
About task scope
Creating a task
Starting a task manually
Starting a task for selected devices
Viewing the task list
General task settings
Exporting a task
Importing a task
Starting the Change tasks password wizard
Step 1. Specifying credentials
Step 2. Selecting an action to take
Step 3. Viewing the results
Viewing task run results stored on the Administration Server
Manual setup of the group task for scanning a device with Kaspersky Endpoint Security
Managing detection and response tasks
Viewing the tasks table
Viewing task run results
Restarting a task
Application tags
Creating an application tag
Renaming an application tag
Assigning tags to an application
Removing assigned tags from an application
Deleting an application tag
Granting offline access to the external device blocked by Device Control
Registering Kaspersky Industrial CyberSecurity for Networks application in OSMP Console
Managing users and user roles
About user accounts
About user roles
Configuring access rights to application features. Role-based access control
Access rights to application features
Predefined user roles
Assigning access rights to specific objects
Assigning permissions to users and groups
Adding an account of an internal user
Creating a security group
Editing an account of an internal user
Changing a user work status
Editing a security group
Assigning a role to a user or a security group
Adding user accounts to an internal security group

Assigning a user as a device owner
Configuring two-factor authentication
About two-factor authentication for an account
Configuring a two-factor authentication secret key at OSMP Console sign-in
Allowing users to configure two-factor authentication
Resetting and deleting a two-factor authentication secret key
Excluding accounts from two-factor authentication
Editing the name of a security code issuer
Changing the number of allowed password entry attempts
Deleting a user or a security group
Changing the password for a user account
Creating a user role
Editing a user role
Editing the scope of a user role
Deleting a user role
Associating policy profiles with roles
Configuring Kerberos domain authentication
Updating Kaspersky databases and applications
Scenario: Regular updating Kaspersky databases and applications
About updating Kaspersky databases, software modules, and applications
Creating the Download updates to the Administration Server repository task
Viewing downloaded updates
Verifying downloaded updates
Creating the task for downloading updates to the repositories of distribution points
Adding sources of updates for the Download updates to the Administration Server repository task
Approving and declining software updates
Automatic installation of updates for Kaspersky Endpoint Security for Windows
About using diff files for updating Kaspersky databases and software modules
Enabling the Downloading diff files feature
Downloading updates by distribution points
Updating Kaspersky databases and software modules on offline devices
Remote diagnostics of client devices
Opening the remote diagnostics window
Enabling and disabling tracing for applications
Downloading trace files of an application
Deleting trace files
Downloading application settings
Downloading system information from a client device
Downloading event logs
Starting, stopping, restarting the application
Running the remote diagnostics of Kaspersky Security Center Network Agent and downloading the results
Running an application on a client device
Generating a dump file for an application
Running remote diagnostics on a Linux-based client device
Managing third-party applications and executable files on client devices
Using Application Control to manage executable files
Application Control modes and categories

Obtaining and viewing a list of applications installed on client devices
Obtaining and viewing a list of executable files stored on client devices
Creating an application category with content added manually
Creating an application category that includes executable files from selected devices
Creating an application category that includes executable files from selected folder
Viewing the list of application categories
Configuring Application Control in the Kaspersky Endpoint Security for Windows policy
Adding event-related executable files to the application category
Creating the Install required updates and fix vulnerabilities task
Find vulnerabilities and required updates task settings
About the license
OpenAPI Reference Guide
Monitoring, reporting, and audit
Scenario: Monitoring and reporting
About types of monitoring and reporting
Triggering of rules in Smart Training mode
Viewing and confirming detections performed using Adaptive Anomaly Control rules
Adding exclusions from the Adaptive Anomaly Control rules
Dashboard and widgets
Using the dashboard
Administration and protection widgets
Adding widgets to the dashboard
Hiding a widget from the dashboard
Moving a widget on the dashboard
Changing the widget size or appearance
Changing widget settings
Detection and response widgets
Basics of managing widgets
Special considerations for displaying data in widgets
Creating a widget
Editing a widget
Deleting a widget
Widget settings
Events widget
Active lists widget
Context tables widget
Assets customized widget
Other widgets
Filtering widget data
Filtering data in a single widget
Filtering data in multiple widgets
Drilling down to a related dashboard
Displaying tenant names in Active list type widgets
Detection and response dashboard
Creating a dashboard layout
Selecting a dashboard layout
Selecting a dashboard layout as the default

- Editing a dashboard layout
- Deleting a dashboard layout
- Enabling and disabling TV mode
- Preconfigured dashboard layouts
- About the Dashboard-only mode
- Configuring the Dashboard-only mode
- Administration and protection reports
- Using reports
- Report template
- Creating a report template
- Viewing and editing report template properties
- Deleting report templates
- Universal report templates
- Exporting a report to a file
- Generating and viewing a report
- Creating a report delivery task
- Detection and response reports
- Report template
- Creating report template
- Configuring report schedule
- Editing report template
- Copying report template
- Deleting report template
- Exporting or importing a report template
- Universal report templates
- Generated reports
- Viewing reports
- Generating reports
- Saving reports
- Updating reports
- Deleting reports
- Events and event selections
- About events in Open Single Management Platform
- Events of Open Single Management Platform components
- Data structure of event type description
- Administration Server events
- Administration Server critical events
- Administration Server functional failure events
- Administration Server warning events
- Administration Server informational events
- Network Agent events
- Network Agent warning events
- Network Agent informational events
- Using event selections
- Creating an event selection
- Editing an event selection
- Viewing a list of an event selection

Exporting an event selection
Importing an event selection
Viewing details of an event
Exporting events to a file
Viewing an object history from an event
Deleting events
Deleting event selections
Setting the storage term for an event
Blocking frequent events
About blocking frequent events
Managing frequent events blocking
Removing blocking of frequent events
Event processing and storage on the Administration Server
Notifications and device statuses
Using notifications
Viewing onscreen notifications
About device statuses
Configuring the switching of device statuses
Configuring notification delivery
Testing notifications
Event notifications displayed by running an executable file
Monitoring event sources
Source status
List of event sources
Monitoring policies
Kaspersky announcements
About Kaspersky announcements
Specifying Kaspersky announcements settings
Disabling Kaspersky announcements
Cloud Discovery
Enabling Cloud Discovery by using the widget
Adding the Cloud Discovery widget to the dashboard
Viewing information about the use of cloud services
Risk level of a cloud service
Blocking access to unwanted cloud services
Exporting events to SIEM systems
Configuring event export to SIEM systems
Before you begin
About event export
About configuring event export in a SIEM system
Marking of events for export to SIEM systems in Syslog format
About marking events for export to SIEM system in the Syslog format
Marking events of a Kaspersky application for export in the Syslog format
Marking general events for export in Syslog format
About exporting events using Syslog format
Configuring Open Single Management Platform for export of events to a SIEM system

Exporting events directly from the database
Executing an SQL query by using the klsql2 utility
Example of an SQL query in the klsql2 utility
Viewing the Open Single Management Platform database name
Viewing export results
Managing object revisions
Viewing and saving a policy revision
Rolling back an object to a previous revision
Deletion of objects
Downloading and deleting files from Quarantine and File backups
Downloading files from Quarantine and File backups
About removing objects from the Quarantine, File backups, or Active threats repositories
Uploading files for scan
Scanning files in the File repository manually
Viewing scan status and results
Multitenancy
About binding Administration Servers to tenants
Configuring integration with Open Single Management Platform
Viewing and editing tenants
Adding new tenants
Assigning roles to users and groups in a tenant
Deleting tenants
Configuring email notifications
Creating a notification template
Editing and deleting a notification template
Configuring the storing of credentials to access the shared network resources
Configuring IOC scanning on schedule
Contact Technical Support
How to get technical support
Technical support via Kaspersky CompanyAccount
Known issues
Troubleshooting
Appendices
Commands for components manual starting and installing
Integrity check of KUMA files
Normalized event data model
Configuring the data model of a normalized event from KATA EDR
Asset data model
User account data model
Audit events
The user has edited the set of fields settings to define sources
Key file added
Reserve license key added
Reserve license key deleted
Application activated with an activation code
License term extended
Invalid activation code

License code blocked
License key deleted
EPS exceeded
License expiration and start of grace period
End of grace period
Expired license key added
Changed the set of spaces to differentiate access to events
Service monitoring thresholds changed
KUMA Core settings modified
GeoIP databases imported
Service was created
Service was deleted
Service was reloaded
Service was restarted
Service was started
Service was paired
Service status was changed
Partition was deleted by user
Partition automatically deleted after retention period is over
Partition deleted by absolute retention period
Partition deleted by relative retention period
Partition moved to cold storage
Error occurred while moving partition to cold storage
Active list was successfully cleared or operation failed
Active list item was successfully changed, or operation was unsuccessful
Active list item was successfully deleted or operation failed
Active list was successfully imported or operation failed
Active list was exported
Context table exported
Context table successfully imported or operation failed
Context table item successfully modified or operation failed
Context table item successfully deleted or operation failed
Context table successfully cleared or operation failed
Resource was added
Resource was deleted
Resource was updated
Importing resources
Settings were updated
The dictionary was successfully updated on the service or operation failed
Dictionary entry added
Dictionary entry successfully deleted or the operation failed
Extended event schema field created
Extended event schema field edited
Extended event schema field imported
Normalizer with extended event schema field imported
Extended event schema field deleted
Request sent to KIRA

KICS/KATA response
Kaspersky Automated Security Awareness Platform response
KEDR response
Importing MITRE ATT&CK techniques and tactics
Metrics alert registered for service or operation failed
Response in Active Directory
Backing up partitions, restoring from backup, importing partitions from an external KUMA instance
Natural language query sent for conversion to SQL query by using KIRA
Correlation rules
Time format
Mapping fields of predefined normalizers
Glossary
Administrator host
Agent
Alert
Asset
Bootstrap
Collector
Configuration file
Context
Correlation rule
Correlator
Custom actions
Distribution package
EPP application
Event
Event router
Forensics
Incident
Investigation graph
Kaspersky Deployment Toolkit
Kubernetes cluster
KUMA inventory file
KUMA services
Lightweight Nagent (LWNGT)
Multitenancy
Network Agent
Network Location Awareness (NLA)
Node
Normalized event
Observables
Playbook
Playbook algorithm
Registry
Response actions
Segmentation rules
Storage

- Target hosts
- Tenant
- Threat development chain
- Transport archive

Information about third-party code

Trademark notices

The MITRE Corporation Terms of Use

Kaspersky Next XDR Expert

Kaspersky Next XDR Expert is a complex solution for the cybersecurity of your business, and it includes Kaspersky applications aimed to protect your devices and infrastructure from cybersecurity risks, and to track and respond to most cyberthreats. The Kaspersky Next XDR Expert components are deployed on a single platform called Open Single Management Platform. The platform provides you with a single user interface for cross-application scenarios, and enables you to integrate both Kaspersky and third-party applications into a unified protection system.

One of the key components of the solution is a SIEM system that allows you to track the events from all components and to perform mutual correlation of the events by using both preset and custom rules. By analyzing logs and telemetry data from the organization's infrastructure, Kaspersky Next XDR Expert automatically detects attacks and allows you to investigate incidents by using a unified investigation graph that combines all events received by Kaspersky Next XDR Expert, from both Kaspersky and third-party applications.

To respond to complex incidents, Kaspersky Next XDR Expert uses both predefined and custom playbooks. Also, the response options include the response actions from third-party applications and complex response actions implemented through several applications.

The solution comprises basic protection of endpoint devices that allows you to block attacks on the endpoint device infrastructure, including both physical and virtual devices. Moreover, the Kaspersky Next XDR Expert components provide specialized protection of mail servers and incoming and outgoing email messages against harmful objects, spam, and phishing.

With Kaspersky Next XDR Expert, you can perform centralized deployment of Kaspersky security applications on the infrastructure devices, run the virus scan tasks and the update tasks remotely, as well as configure the security policies of the managed applications. The monitoring dashboard displays the current protection system state, detailed reports, and policy parameters.

Kaspersky Next XDR Expert components

	Open Single Management Platform		Kaspersky Industrial CyberSecurity for Networks
	Kaspersky Unified Monitoring and Analysis Platform		Kaspersky Industrial CyberSecurity for Nodes
	Kaspersky Endpoint Detection and Response		Kaspersky CyberTrace
	Kaspersky Anti-Targeted Attack Platform		Kaspersky Automated Security Awareness Platform
	Kaspersky Security for Linux Mail Server		Kaspersky Hybrid Cloud Security
	Kaspersky Endpoint Security for Business		

Quick links

New features

- What's new in Kaspersky Next XDR Expert

Key features

- Managing alerts and security incidents
- Threat hunting tools
- Investigation graph
- Predefined and custom playbooks
- Manual threat response actions
- Dashboard and widgets

Compatibility and hardware and software requirements

- Hardware and software requirements
- Compatible applications and solutions
- Integration with other solutions and third-party systems

Getting started

- Walk-through scenario of deployment, activation and initial configuration of Kaspersky Next XDR Expert
- Deployment of Kaspersky Next XDR Expert
- Migration to Kaspersky Next XDR Expert
- Using the threat monitoring, detection and hunting capabilities
- Example of incident investigation with Kaspersky Next XDR Expert

Working with Open Single Management Platform

- Installing Kaspersky security applications on devices on a corporate network
- Remotely run scan and update tasks
- Managing the security policies of managed applications

What's new

Kaspersky Next XDR Expert 2.0

Kaspersky Next XDR Expert has several new features and improvements:

- Unified OSMP Console: You can now manage scenarios of Kaspersky Unified Monitoring and Analysis Platform and Kaspersky Endpoint Detection and Response scenarios in Kaspersky Next XDR Expert.
- New response actions from Kaspersky Endpoint Detection and Response are now available:
  - Managing services
  - Deleting a file
  - Getting a file
  - Getting a file from quarantine
  - Scanning files obtained from devices or uploaded by users
  - Getting memory dump
  - Getting a process memory dump
  - Getting a disk image
  - Getting forensics
  - Getting a registry key
  - Getting NTFS metafiles
  - YARA scan
  - IOC scan
  - Restoring files from quarantine
  - Deleting files from quarantine
- New AI-powered features:
  - Generate a brief report for alerts and incidents with KIRA to improve analysis efficiency.
  - AI analyzes account behavior and generates alerts when it detects suspicious activity or signs of account compromise.
  - AI detects attacks where a legitimate application loads a malicious DLL that replaces the original.
  - AI can now help you generate SQL queries in the **Threat hunting** section.
- Two-factor authentication is now required for all accounts to enhance security.

- You can now launch playbooks in test mode to check their functionality before execution.
- You can now also launch response actions from the **Assets (Devices)** and **Response history** sections, and from **Event details**.
- Response actions can now be rerun with updated parameters.
- Added remote terminal access for direct device investigation and response actions.
- Investigation graph improvements:
  - You can now place events on the investigation graph according to the timeline. Earlier events appear on the left, later ones on the right.
  - Event trees are now displayed in a dedicated panel instead of directly on the graph.
  - Events on the graph are now arranged along a timeline: earlier events appear on the left, later ones on the right.
  - You can now add comments both to the entire process tree and to individual nodes.
  - When zooming the graph, nodes can be grouped into clusters, which you can expand to view the underlying nodes at an optimized scale.
- Implemented a new data model for events.
- New alert types have been added:
  - Alerts generated based on file scan results.
  - Rescan alerts are created when files are scanned again after not being analyzed in a previous alert.
  - Alerts generated by TAA (IOA) detection technology.
  - Alerts generated by IOC detection technology.
  - Alerts generated by YARA detection technology.
- IOA rule exclusion settings can now be configured in Kaspersky Endpoint Security for Windows policies, either directly in the policy or from alert or event details.
- You can now configure user authentication policies, including password lifetime and the allowed inactivity period before a user session is terminated.
- Support for hierarchical multitenancy has been added. Installations can be organized into parent and child instances, with the ability to connect to child instances from a parent instance.
- The **Assets (Devices)** section now includes two tabs:
  - **Assets (Devices)**, where you can manage and categorize devices.
  - **Analysis and response**, where you can launch response actions for listed assets and connect to them via a remote terminal.

- You can now create and manage the following custom rules:
  - Prevention rules
  - IOC rules
  - YARA rules
  - Sandbox rules
- Automatic and manual file sending to Sandbox is now available from Kaspersky Endpoint Security for Windows and Kaspersky Endpoint Security for Linux when integration with Kaspersky Sandbox is enabled.
- Air-gapped deployment is now available:
  - Prepare a package archive on the administrator host with internet access.
  - Transfer the package archive to an isolated environment.
  - Install Kaspersky Next XDR Expert on target hosts without internet access.
- Added support for uploading server and Endpoint Agent certificates to establish secure connections between them.
- Added support for uploading certificates to connect to Sandbox.
- FreeIPA and ALD Pro domain controllers are now supported.
- Your account menu is now moved from the main menu to the upper pane.
- New language support: Kaspersky Next XDR Expert is now available in the Chinese Simplified and Chinese Traditional languages.
- New features and improvements introduced in the SIEM component (KUMA):
  - The Metrics service can be installed on an arbitrary host. Only one Metrica service must exist. When installing KUMA from scratch, you must specify the host for installing the Metrics service in the `kuma_metrics` section of the inventory file. When upgrading KUMA to version 4.2, you must specify the host on which the active Metrics service is installed in the `kuma_metrics` section of the inventory file. For example, if in KUMA 4.0 the Metrics service was installed on the host kuma-core.example.com, before upgrading to version 4.2, you need to specify the kuma-core.example.com host in the kuma_metrics section of the inventory file. If you want to migrate the Metrics service to another host, you can do so using the expand.sh installer and the expand.inventory.yml inventory file.

    When migrating the KUMA Core to a Kubernetes cluster, you must first migrate the Metrics service to the same host as the KUMA Core using the expand.sh installer and the expand.inventory.yml inventory file. When migrating the KUMA Core to a Kubernetes cluster, a check is made to see if the Metrics service is running on this host. If the Metrics service is not found, an error is returned and the migration is stopped. When you migrate the KUMA Core to a Kubernetes cluster, metrics data is migrated to the cluster.
  - You can back up partitions on a schedule, back up partitions manually, restore partitions from a backup, or import partitions from an external KUMA installation.
  - Now you can back up and restore the KUMA Core using the web interface.
  - For cold storage, data from the ClickHouse cluster can now be transferred to S3 cloud storage.

- The kafka connector now uses the SCRAM-SHA-256 and SCRAM-SHA-512 secure connection mechanisms for authorization.
- Introducing the universalAPI connector for getting data from arbitrary REST APIs with JSON support, filtering by ID, and iteration mode.
- Now you can run SQL queries in the background with a lower priority and view the results after execution is complete.
- Under a General Administrator account, you can now view and manage active requests of other users.
- Introducing the periodical correlation rule type, which is used to detect unauthorized use of accounts by detecting anomalous user activity that may indicate a compromised account. This rule can be created only when using the correlator-ng correlator.
- Now you can configure the time zone and ISO 8601 format in the user account, as well as customize the first day of the week for the entire KUMA instance (available only to General Administrators).
- You can now link resources (correlation rules, enrichment rules, response rules, aggregation rules, and destinations) to services such as collectors, correlators, and event routers. The available services depend on the type of the selected resource.
- A new type of destinations, udp, allows normalized events to be transmitted from collectors and correlators via UDP.
- In context tables, you can now export data in CSV and TSV formats.

- New features and improvements introduced in Kaspersky Security Center for Linux 16.1.

Kaspersky Next XDR Expert 1.4

Kaspersky Next XDR Expert has several new features and improvements:

- When upgrading Kaspersky Next XDR Expert, you may need to update Bootstrap depending on the version you are upgrading from.
  - **Upgrades that require a Bootstrap update**

    Update Bootstrap for the following upgrades of Kaspersky Next XDR Expert:

    - Version 1.3 → Version 1.4 Hotfix 6 or later
    - Version 1.4 → Version 1.4 Hotfix 6 or later
    - Version 1.4 Hotfix 1 → Version 1.4 Hotfix 6 or later
    - Version 1.4 Hotfix 2 → Version 1.4 Hotfix 6 or later

    To update Bootstrap, run the following command:

    ```
    ./kdt apply -k <path_to_XDR_updates_archive> -i <path_to_configuration_file>
    --force-bootstrap
    ```

    After that, upgrade Kaspersky Next XDR Expert by running the following command:

```
./kdt apply -k <path_to_XDR_updates_archive>
```

- **Upgrades that do not require a Bootstrap update**

  Do not update Bootstrap for the following upgrades of Kaspersky Next XDR Expert:

  - Version 1.4 Hotfix 3 → Version 1.4 Hotfix 6 or later
  - Version 1.4 Hotfix 4 → Version 1.4 Hotfix 6 or later

  To upgrade Kaspersky Next XDR Expert, run the following command:

  ```
  ./kdt apply -k <path_to_XDR_updates_archive>
  ```

- You can now set a fixed response time (SLA) for incident handling. SLA metrics include the time to resolve an incident fully, the time to first response, and the time to confirm the incident.
- Postgres Pro Built-in High Availability cluster is now supported. The nodes of this DBMS cluster are specified in the connection string when installing Kaspersky Next XDR Expert.
- You can now save a backup copy of Kaspersky Next XDR Expert data, and then restore data from the created copy.
- You can now add or delete worker nodes included in the Kubernetes cluster, without restarting the cluster nodes.
- Saving a playbook as a draft is supported. The playbook draft cannot be launched, but it is available for editing in the tenant to which the playbook belongs. To launch the playbook draft, you need to publish it first.
- You can now leave comments for any graph node or link.
- Exporting incidents to PDF is supported.
- You can now export to a CSV file information about assets affected by or involved in the alert or incident, as well as information about observables that relate to the alerts linked to the current incident.
- You can assign alerts and incidents to a user group manually or by using a playbook. If you want alerts and incidents to be automatically distributed among users of the group based on their workload and availability status, you can enable automatic assignment.
- The playbook visual editor is now used to create playbooks. The playbook visual editor allows you to build and visualize the playbook execution flow and the relations between the playbook algorithm steps.
- Names of an alert and incident are displayed when linking the alert to an incident. This allows filtering alerts and incidents by their names.
- Filtering of alerts and incidents is improved.
- If multiple aggregation rules are applied to the alert, these aggregation rules are now processed sequentially according to the specified priority, not just the first rule found.
- Connection profiles for out-of-office users with Linux devices are now available. By using connection profiles, you can configure the rules for Network Agents on Linux devices to connect to the same or different Administration Servers, depending on the device location.

- Transmitting encryption keys between Administration Servers is now available.
- The **Scan** button is deleted from the **Quarantine** section.

Kaspersky Next XDR Expert 1.3

Kaspersky Next XDR Expert has several new features and improvements:

- An updated version of Bootstrap is used in the application. Before you install the new version of Kaspersky Next XDR Expert, update Bootstrap by running the following command:

  ```
  ./kdt apply -k <path_to_XDR_updates_archive> -i <path_to_configuration_file> --force-bootstrap
  ```

- You can now create a child incident. Child incidents allow you to investigate and respond to incidents across different tenants.
- Kaspersky NGFW is supported for incident response scenarios. You can create the services for Kaspersky NGFW by using the KUMA inventory file. Kaspersky NGFW includes the following features:
  - Blocking IP addresses through Kaspersky NGFW.
  - New dashboards for Kaspersky NGFW.
- You can now change the Kubernetes cluster IP address and FQDN of services by using Kaspersky Deployment Toolkit without reinstalling.
- Pre-validation of parameters in the configuration file. Kaspersky Deployment Toolkit verifies the presence and values of parameters in the configuration file before starting the installation or update of Kaspersky Next XDR Expert. The feature is available for both single-node and multi-node deployments.
- You can now manage aggregation rules in the user interface of Kaspersky Next XDR Expert.
- A new widget, **Incidents by type**, is added.
- The following dashboard widgets now display information on unclosed incidents:
  - Active incidents
  - Active incidents by tenant
  - Incidents by severity
  - Affected assets in incidents
  - Affected asset categories in incidents
  - Latest incidents
- The limit on the number of alerts linked to incidents that you can merge has been increased. The maximum number of alerts linked to incidents for merging is 1000. When merging incidents, the number of alerts linked to these incidents is displayed in the **Merge incidents** window.

- Private IP addresses are no longer included in the observables related to the alert or incident.
- Kaspersky Next XDR Expert now supports new versions of EPP applications.

Kaspersky Next XDR Expert 1.2

Kaspersky Next XDR Expert has several new features and improvements:

- An updated version of Bootstrap is used in the application. Before you install the new version of Kaspersky Next XDR Expert, update Bootstrap by running the following command:

  ```
  ./kdt apply -k <path_to_XDR_updates_archive> -i <path_to_configuration_file> --force-bootstrap
  ```

- Kaspersky Next XDR Expert upgrade from version 1.1 to version 1.2.
- Optimized Kaspersky Next XDR Expert deployment: improved configuration file and Configuration wizard for a simplified specifying of the installation parameters.
- Deployment preliminary checks. Before you deploy Kaspersky Next XDR Expert, you can now check if the system requirements are met. Kaspersky Deployment Toolkit (KDT) checks your hardware, operating system, software, and network environment. If at least one requirement is not met, KDT interrupts the deployment and provides you a detailed report.
- Flexible incident workflow. You can configure an incident workflow and view it in the visual editor.
- You can now attach files to alerts or incidents. If necessary, you can remove or download the attached files.
- Customizable incident handling process by using incident types.
- When creating a playbook, you can configure the playbook algorithm to edit the incident properties or the alert properties.
- You can export information about all incidents displayed in the incident table to a JSON file. This may be required when you have to provide this information to third parties.
- AI-based asset scoring. A machine learning-based engine helps you evaluate the processes running on an asset, and define if a particular process is normal or if it is unusual and requires attention from a SOC analyst.
- Improved the configuration process of the templates for email notifications about events occurring in Kaspersky Next XDR Expert.
- You can reduce or increase the retention periods of alerts and incidents, depending on your needs. By default, the retention period of alerts and incidents is 360 days.
- Uninstallation of Kaspersky Next XDR Expert. All created data will also be removed.
- From a shortcut menu in the alert details window or incident details window, you can now open the **Threat hunting** page on a new browser tab.
- In the alert details window or incident details window, you can now search through affected assets and observables.
- Ability to configure alert aggregation rules through the REST API.

- When you open the **Threat hunting** page from the alert details window or incident details window, the search is now performed for the period between the first and the last event of the alert or incident, and not for the last 24 hours.

- Deployment preliminary checks. Before you deploy Kaspersky Next XDR Expert, you can now check if the system requirements are met. Kaspersky Deployment Toolkit (KDT) checks your hardware, operating system, software, and network environment. If at least one requirement is not met, KDT interrupts the deployment and provides you a detailed report.

- Open Single Management Platform can now be installed on the Nutanix AHV virtualization platform.

- OSMP Console optimization: the console windows, login page, and the Dashboard now load faster.

- You can now switch from the incident details window to the incident-related events on the **Threat hunting** page.

- Kaspersky Next XDR Expert now supports the following EPP applications:

  - Kaspersky Endpoint Security for Windows, versions 12.5, 12.6, 12.7

  - Kaspersky Endpoint Security 12.1 for Linux

  - Kaspersky Endpoint Security 12.1 for Mac

  - Kaspersky Industrial CyberSecurity for Nodes 4.0

  - Kaspersky Endpoint Agent 4.0

- Kaspersky Next XDR Expert is now compatible with Kaspersky Anti Targeted Attack Platform 7.0.

- You can now refresh the information in the alert details window and the incident details window by clicking the refresh icon.

Kaspersky Next XDR Expert 1.1

Kaspersky Next XDR Expert has several new features and improvements:

- An updated version of Bootstrap is used in the application. Before you install the new version of Kaspersky Next XDR Expert, update Bootstrap by running the following command:

  ```
  ./kdt apply -k <path_to_XDR_updates_archive> -i <path_to_configuration_file> --force-bootstrap
  ```

- New design of the user interface.

- Reduced hardware and software requirements.

- Increased application stability.

- A new deployment wizard for the simplified configuration of the installation parameters.

- Addition of predefined playbooks.

- Kaspersky Next XDR Expert now supports the following EPP-applications:
  - Kaspersky Endpoint Security 12.0 for Mac
  - Kaspersky Industrial CyberSecurity for Nodes 3.2
  - Kaspersky Endpoint Agent 3.16
- New Dashboard widgets for monitoring responses performed through playbooks.
- Migration from Kaspersky Security Center to Kaspersky Next XDR Expert, including migration of users and tenants, and the binding of tenants to Administration Servers of Kaspersky Security Center.
- Kaspersky Next XDR Expert is now compatible with Kaspersky Anti Targeted Attack Platform 6.0.
- New features and improvements introduced in the August 2024 update of Kaspersky Unified Monitoring and Analysis Platform.

About Kaspersky Next XDR Expert

Kaspersky Next XDR Expert (XDR) is a robust cybersecurity solution that defends your corporate IT infrastructure against sophisticated cyberthreats, including those that cannot be detected by EPP applications installed on corporate assets. It provides full visibility, correlation, and automation; and leverages a diverse range of response tools and data sources, including endpoint assets, and network and cloud data. To protect your IT infrastructure effectively, Kaspersky Next XDR Expert analyzes the data from these sources to identify threats, create alerts for potential incidents, and provide the tools to respond to them. Kaspersky XDR is backed by advanced analytics capabilities and a strong track record of security expertise.

This solution provides a unified detection and response process through integrated components and holistic scenarios in a single interface to improve the efficiency of security professionals.

The detection tools include:

- Threat hunting tools to proactively search for threats and vulnerabilities by analyzing events.
- Advanced threat detection and cross-correlation: real-time correlation of events from different sources, out-of-the-box correlation rules for various scenarios with MITRE ATT&CK matrix mapping, ability to create new rules and customize existing ones, and retrospective scans for detecting zero-day vulnerabilities.
- An investigation graph to visualize and facilitate an incident investigation and identify the root causes of the alert.
- Use of Kaspersky Threat Intelligence Portal to get the latest detailed threat intelligence, for example, about web addresses, domains, IP addresses, file hashes, statistical and behavioral data, and WHOIS and DNS data.

The response tools include:

- Manual response actions: asset isolation, run commands, add and delete prevention rules, launch tasks on an asset, Kaspersky Threat Intelligence Portal reputation enrichment, and training assignments for users.
- Playbooks, both predefined and user-created, to automate typical response operations.
- Third-party application response actions and cross-application response scenarios.

Kaspersky Next XDR Expert includes the capabilities of endpoint threat detection and response (from Kaspersky Endpoint Detection and Response):

- Comprehensive view of endpoints across the entire corporate infrastructure.
- Detection of threats based on indicators of compromise (IOC), YARA rules, and unique indicators of attack (IOA).
- Sandbox for the safe analysis of suspicious files and programs, providing information on object behavior in an isolated environment.

Kaspersky Next XDR Expert also takes advantage of the Open Single Management Platform component for asset management and the centralized run of security administration and maintenance tasks:

- Deploying Kaspersky applications on the assets in the corporate network.
- Remotely launching scan and update tasks.

- Obtaining detailed information about asset protection.
- Configuring all the security components by using Kaspersky applications.

Kaspersky Next XDR Expert supports the hierarchy of tenants.

Kaspersky Next XDR Expert is integrated with Active Directory, includes APIs, and supports a wide range of integrations both with Kaspersky applications and third-party solutions for data obtaining and responding. For information about the applications and solutions that Kaspersky Next XDR Expert supports, see the Compatible Kaspersky applications and Integration with other solutions sections.

> Updates functionality (including providing anti-virus signature updates and codebase updates), as well as KSN functionality may not be available in the software in the U.S.

Hardware and software requirements

This article describes hardware requirements of single-node deployment scheme and multi-node deployment scheme, software requirements of Open Single Management Platform (including Kaspersky Endpoint Detection and Response functionality), hardware and software requirements of OSMP components.

For hardware requirements of Kaspersky EDR Expert, refer to Kaspersky Endpoint Detection and Response Expert Help.

Single-node deployment: hardware requirements

Single-node deployment requires less resources (see the table below), but the following considerations should be taken into account:

- Single-node scheme only supports up to 10,000 devices in the network.
- The database is located on the primary worker node outside the Kubernetes cluster.

  In case of single-node deployment, it is strongly recommended that you first install the DBMS manually on the host that will act as a primary node. After that, you can deploy Kaspersky Next XDR Expert on the same host.
- Sandbox is located on a physical server outside the Kubernetes cluster.
- To deploy the solution correctly, ensure that CPU of the target host supports the BMI, AVX, and SSE 4.2 instruction set.

Minimum hardware requirements

Hardware requirements for a single-node deployment scheme

Solution	250 devices	1000 devices	3000 devices	5000 devices	10,000 devices

A solution that includes the following components:
• Open Single Management Platform
• Kaspersky Unified Monitoring and Analysis Platform
• Sandbox
	1 XDR primary worker node:
	• CPU: 10 cores, operating frequency of 2.5 GHz
	• RAM: 44 GB
	• Available SSD disk space:
	  • First disk subsystem: 400 GB
	  • Secondary disk subsystem: 500 GB
	1 Sandbox node:
	• CPU: 32 cores
	• RAM: 80 GB
	• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
		1 XDR primary worker node:
		• CPU: 16 cores, operating frequency of 2.5 GHz
		• RAM: 48 GB
		• Available SSD disk space:
		  • First disk subsystem: 400 GB
		  • Secondary disk subsystem: 600 GB
		1 Sandbox node:
		• CPU: 32 cores
		• RAM: 80 GB
		• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
			1 XDR primary worker node:
			• CPU: 22 cores, operating frequency of 2.5 GHz
			• RAM: 54 GB
			• Available SSD disk space:
			  • First disk subsystem: 600 GB
			  • Secondary disk subsystem: 1 TB
			1 Sandbox node:
			• CPU: 32 cores
			• RAM: 80 GB
			• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
				1 XDR primary worker node:
				• CPU: 32 cores, operating frequency of 2.5 GHz
				• RAM: 62 GB
				• Available SSD disk space:
				  • First disk subsystem: 800 GB
				  • Secondary disk subsystem: 1.4 TB
				1 Sandbox node:
				• CPU: 32 cores
				• RAM: 80 GB
				• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
					1 XDR primary worker node:
					• CPU: 36 cores, operating frequency of 2.5 GHz
					• RAM: 74 GB
					• Available SSD disk space:
					  • First disk subsystem: 1.5 TB
					  • Secondary disk subsystem: 2.4 TB
					1 Sandbox node:
					• CPU: 32 cores
					• RAM: 80 GB
					• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.

Multi-node deployment: hardware requirements

Multi-node deployment requires more resources (see the table below). For this scheme, the following considerations should be taken into account:

- Multi-node cluster scheme is recommended for networks that exceed 10,000 devices.
- The database is located on a separate host outside the cluster. To support a high-availability database configuration, use at least three database nodes.
- Sandbox is located on a physical server outside the Kubernetes cluster.
- To deploy the solution correctly, ensure that CPUs of target hosts support the BMI, AVX, and SSE 4.2 instruction set.

Minimum hardware requirements

Hardware requirements for a multi-node deployment scheme

Solution	20,000 devices	30,000 devices	50,000 devices
A solution that includes the following components: • Open Single Management Platform • Kaspersky Unified Monitoring and Analysis Platform • Sandbox	12 nodes: • 1 XDR primary node • 3 XDR worker nodes • 1 XDR database node • 1 KUMA collector • 1 KUMA correlator • 3 KUMA keeper • 2 KUMA storage	12 nodes: • 1 XDR primary node • 3 XDR worker nodes • 1 XDR database node • 1 KUMA collector • 1 KUMA correlator • 3 KUMA keeper • 2 KUMA storage	12 nodes: • 1 XDR primary node • 3 XDR worker nodes • 1 XDR database node • 1 KUMA collector • 1 KUMA correlator • 3 KUMA keeper • 2 KUMA storage

	1 XDR primary node:
	• CPU: 4 cores
	• RAM: 8 GB
	• Available SSD disk space: 500 GB
	3 XDR worker nodes:
	• CPU: 10 cores
	• RAM: 20 GB
	• Available SSD disk space: 1 TB
	1 XDR database node:
	• CPU: 10 cores
	• RAM: 22 GB
	• Available SSD disk space: 1.6 TB
	1 KUMA collector node:
	• CPU: 8 cores
	• RAM: 16 GB
	• Available SSD disk space: 500 GB
	1 KUMA correlator node:
	• CPU: 8 cores
	• RAM: 32 GB
	• Available SSD disk space: 500 GB
	3 KUMA keeper nodes:
	• CPU: 6 cores
	• RAM: 12 GB
	• Available SSD disk space: 200 GB
	2 KUMA storage nodes:
	• CPU: 24 cores
	• RAM: 64 GB
	• Available disk space:
	  • Hot storage (SSD disk): 4.7 TB
	  • Cold storage (HDD disk): 15.7 TB
	1 Sandbox node:
	• CPU: 32 cores
	• RAM: 80 GB
	• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
	Network bandwidth: 10 Gbps for each node
		1 XDR primary node:
		• CPU: 4 cores
		• RAM: 8 GB
		• Available SSD disk space: 500 GB
		3 XDR worker nodes:
		• CPU: 12 cores
		• RAM: 24 GB
		• Available SSD disk space: 1 TB
		1 XDR database node:
		• CPU: 12 cores
		• RAM: 24 GB
		• Available SSD disk space: 2.7 TB
		1 KUMA collector node:
		• CPU: 8 cores
		• RAM: 16 GB
		• Available SSD disk space: 500 GB
		1 KUMA correlator node:
		• CPU: 8 cores
		• RAM: 32 GB
		• Available SSD disk space: 500 GB
		3 KUMA keeper nodes:
		• CPU: 6 cores
		• RAM: 12 GB
		• Available SSD disk space: 200 GB
		2 KUMA storage nodes:
		• CPU: 24 cores
		• RAM: 64 GB
		• Available disk space:
		  • Hot storage (SSD disk): 7.1 TB
		  • Cold storage (HDD disk): 23.6 TB
		1 Sandbox node:
		• CPU: 32 cores
		• RAM: 80 GB
		• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
		Network bandwidth: 10 Gbps for each node
			1 XDR primary node:
			• CPU: 4 cores
			• RAM: 8 GB
			• Available SSD disk space: 500 GB
			3 XDR worker nodes:
			• CPU: 14 cores
			• RAM: 30 GB
			• Available SSD disk space: 1 TB
			1 XDR database node:
			• CPU: 16 cores
			• RAM: 32 GB
			• Available SSD disk space: 4.3 TB
			1 KUMA collector node:
			• CPU: 8 cores
			• RAM: 16 GB
			• Available SSD disk space: 500 GB
			1 KUMA correlator node:
			• CPU: 8 cores
			• RAM: 32 GB
			• Available SSD disk space: 500 GB
			3 KUMA keeper nodes:
			• CPU: 6 cores
			• RAM: 12 GB
			• Available SSD disk space: 200 GB
			2 KUMA storage nodes:
			• CPU: 24 cores
			• RAM: 64 GB
			• Available disk space:
			  • Hot storage (SSD disk): 11.8 TB
			  • Cold storage (HDD disk): 39.3 TB
			1 Sandbox node:
			• CPU: 32 cores
			• RAM: 80 GB
			• Disk subsystem: 2 HDD, 600 GB each, 10,000 RPM.
			Network bandwidth: 10 Gbps for each node

Open Single Management Platform: Software requirements

Software requirements and supported systems and platforms

Operating system	64-bit versions of the following operating systems are supported: Ubuntu Server 22.04 LTS Ubuntu Server 24.04 LTS Debian GNU/Linux 12.x (Bookworm) Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.1) On the target hosts with the Ubuntu family operating systems, the Linux kernel version must be 5.15.0.107 or later.
Packages	Python 3.8 or later
Virtualization platforms	Open Single Management Platform supports all the virtualization platforms listed here. However, there are limitations for Open Single Management Platform components. The differences are presented in the table below.

Virtualization platform name	Open Single Management Platform	KUMA	Sandbox
VMware ESXi 6.7.0	✓	✓	✓
VMware ESXi 7.0	✓	✓	✓
VMWare ESXi 8	✓	✓	—
Nutanix 7.3	✓	✓	—
Kernel-based Virtual Machine (KVM)	✓	✓	—

Open Single Management Platform components

To view the hardware and software requirements for an Open Single Management Platform component, click its name:

- OSMP Console
- Kaspersky Unified Monitoring and Analysis Platform (hereinafter KUMA)
- Secondary Kaspersky Security Center Administration Servers
- Kaspersky Security Center Network Agent
- Kaspersky Endpoint Security for Windows
- Kaspersky Anti Targeted Attack Platform (hereinafter KATA)
- Kaspersky Industrial CyberSecurity for Networks
- Kaspersky Industrial CyberSecurity for Nodes
- Kaspersky CyberTrace
- Kaspersky Threat Intelligence Portal
- Kaspersky Automated Security Awareness Platform (hereinafter KASAP)

Common requirements and considerations

100% vCPU allocation is required if you use virtualization.

For networks that exceed 40,000 devices, use secondary Administration Servers.

Make sure that the DNS server is available on the network.

Single-node deployment cannot be upgraded to multi-node deployment. Multi-node installation is preferable if infrastructure expansion is expected.

Effective device and EPS calculation

Hardware requirements may vary depending on the operating system running on endpoint devices. Use the following formula to estimate effective devices in your network:

```
<number of devices> = <Windows endpoints> + 3* <Linux and macOS endpoints> + 20 *
<servers>
```

An effective device is expected to contribute 0.5 EPS (events per second) with default settings. Total EPS is calculated using the following formula:

```
<total EPS> = <EPS from effective devices> + <third-party EPS>
```

You can convert total EPS to effective devices using the following formula:

```
<total effective devices> = <total EPS> *2
```

Nodes configuration

If IOMMU (Input-Output Memory Management Unit) is supported in BIOS/UEFI, it must be enabled.

All power-saving optimizations in BIOS/UEFI must be disabled, for example, C-States, P-States, Turbo-Boost.

If BIOS/UEFI support CPU performance settings, they must be configured for maximum performance. For more information refer to the hardware vendor documentation.

Disk/Storage Requirements

Minimum input/output performance:

- For single-node deployment : 12,000 IOPS (sequential read/write, 8 KB block size).
- For multi-node deployment : 8,000 IOPS (sequential read/write, 8 KB block size).

The stripe size for RAID arrays must be at least 16 KB.

SSDs/NVMe drives must support TRIM.

The discard mount option is not recommended. Instead, periodic cleanup of unused blocks must be configured to maintain performance.

The fsync delay should not exceed optimal values:

- 2 ms for single-node deployment and control-plane nodes.
- 3 ms for compute nodes with EPS above 1000.
- 10 ms for compute nodes with EPS below 1000.

NUMA (Non-Uniform Memory Access)

On hosts with single-socket or low memory, NUMA should be disabled.

On multi-socket systems, NUMA should be enabled. We recommend that you add at least 20% extra memory to XDR recommended specifications for NUMA overhead.

Kaspersky Deployment Toolkit requirements

All Open Single Management Platform components are installed by using the Kaspersky Deployment Toolkit utility (hereinafter referred to as KDT).

KDT has the following hardware and software requirements:

Specification	System requirements
Hardware	CPU: 4 cores, operating frequency of 2.5 GHz RAM: 8 GB Available disk space: at least 100 GB, including: • Docker data storage directory (default: `/var/lib/docker`): at least 60 GB • Temporary files directory for KDT (default: `/tmp`): at least 15 GB
Operating system	64-bit versions of the following operating systems are supported: • Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.1) • CentOS 7.x • CentOS 8.x • Ubuntu Server 22.04 LTS • Ubuntu Server 24.04 LTS • Debian GNU/Linux 12.x (Bookworm)

Requirements for hosts with KUMA services

The KUMA services (collectors, correlators, and storages) can be installed either on dedicated hosts outside of the Kubernetes cluster or as workloads inside the Kubernetes cluster as part of the Kaspersky Next XDR Expert deployment.

When KUMA services are deployed outside the Kubernetes cluster, the hardware, software, and installation requirements defined in this article apply directly to the hosts where the services are installed.

When KUMA services are deployed inside the Kubernetes cluster, they run on Kubernetes worker nodes and consume cluster resources according to the KUMA hardware, software, and installation requirements defined in this article. In this case, the Kubernetes worker nodes must provide sufficient capacity to satisfy these requirements.

General application installation requirements

You can install the application on the following operating systems:

- Oracle Linux
- Astra Linux
- Ubuntu

  Supported operating system versions are listed in the Hardware and software requirements section.

  Supported configurations are Server and Server with GUI support. In the Server with GUI support configuration, you do not need to install additional packages for reporting.

Before deploying the application, make sure the following conditions are met:

- Servers on which you want to install the components satisfy the hardware and software requirements.
- Ports to be used by the installed instance of KUMA are available.
- KUMA components are addressed using the fully qualified domain name (FQDN) of the host in the hostname.example.com format. Before you install the application, make sure that the correct host FQDN is returned when you run the following command:

  ```
  hostname -f
  ```

- The name of server on which you are running the installer is not `localhost` or `localhost.<domain>`.
- The name of the server on which you are installing KUMA Core does not start with a numeral.
- Time synchronization over Network Time Protocol (NTP) is configured on all servers with KUMA services.

Recommended hardware and software requirements

This section lists the hardware and software requirements for processing a data stream of up to 40,000 events per second (EPS). The KUMA load value depends on the type of events being parsed and the efficiency of the normalizer.

For event processing efficiency, the CPU core count is more important than the clock rate. For example, 8 CPU cores with a medium clock rate can process events more efficiently than 4 CPU cores with a high clock rate. The table below lists the hardware and software requirements of KUMA components.

The amount of RAM utilized by the collector depends on configured enrichment methods (DNS, accounts, assets, enrichment with data from Kaspersky CyberTrace) and whether aggregation is used. RAM consumption is influenced by the data aggregation window setting, the number of fields used for aggregation of data, volume of data in fields being aggregated.

For example, with an event stream of 1000 EPS and event enrichment disabled (event enrichment is disabled, event aggregation is disabled, 5000 accounts, 5000 assets per tenant), one collector requires the following resources:

- 1 CPU core or 1 virtual CPU
- 512 MB of RAM
- 1 GB of disk space (not counting event cache)

For example, to support 5 collectors that do not perform event enrichment, you must allocate the following resources: 5 CPU cores, 2.5 GB of RAM, and 5 GB of free disk space.

Recommended hardware and software requirements for installation of the KUMA services

	Collector	Correlator	Storage
CPU	Intel or AMD with SSE 4.2 support: at least 4 cores/8 threads or 8 virtual CPUs.	Intel or AMD with SSE 4.2 support: at least 4 cores/8 threads or 8 virtual CPUs.	Intel or AMD with SSE 4.2 support: at least 12 cores/24 threads or 24 virtual CPUs.
RAM	16 GB	16 GB	48 GB
Free disk space	/opt directory size: at least 500 GB.	/opt directory size: at least 500 GB.	/opt directory size: at least 500 GB.
Operating systems	• Ubuntu Server 24.04 LTS. • Oracle Linux 8.6, 8.7, 9.2, 9.4. • Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.2).		
Network bandwidth	100 Mbps	100 Mbps	The transfer rate between ClickHouse nodes must be at least 10 Gbps if the data stream exceeds 20,000 EPS.

Installation of KUMA is supported in the following virtual environments:

- VMware 6.5 or later
- Hyper-V for Windows Server 2012 R2 or later
- QEMU-KVM 4.2 or later
- Software package of virtualization tools "Brest" RDTSP.10001-02

Kaspersky recommendations for storage servers

For storage servers Kaspersky specialists recommend the following:

- Put ClickHouse on solid state drives (SSD). SSDs help improve data access speed. Hard drives can be used to store data using the HDFS technology.
- To connect a data storage system to storage servers, use high-speed protocols, such as Fibre Channel or iSCSI 10G. We do not recommend using application-level protocols such as NFS and SMB to connect data storage systems.
- Use the ext4 file system on ClickHouse cluster servers.
- If you are using RAID arrays, use RAID 0 for high performance, or RAID 10 for high performance and fault tolerance.
- To ensure fault tolerance and performance of the data storage subsystem, make sure that ClickHouse nodes are deployed strictly on different disk arrays.
- If you are using a virtualized infrastructure to host system components, deploy ClickHouse cluster nodes on different hypervisors. In this case, it is necessary to prevent two virtual machines with ClickHouse from working on the same hypervisor.
- For high-load KUMA installations, install ClickHouse on physical servers.

Requirements for devices for installing agents

To have data sent to the KUMA collector, you must install agents on the network infrastructure devices. Hardware and software requirements are listed in the table below.

Recommended hardware and software requirements for installation of agents

	Windows devices	Linux devices
CPU	Single-core, 1.4 GHz or higher	Single-core, 1.4 GHz or higher
RAM	512 MB	512 MB
Free disk space	1 GB	1 GB
Operating systems	• Microsoft Windows 2012 • Microsoft Windows Server 2012 R2 • Microsoft Windows Server 2016 • Microsoft Windows Server 2019 • Microsoft Windows 10 20H2, 21H1	• Astra Linux Special Edition RUSB.10015-01 (2023-0426SE17 update 1.7.4) • Ubuntu 22.04 LTS (Jammy Jellyfish) • Debian 11.7 (Bullseye)

Installation requirements for Oracle Linux, Astra Linux, Ubuntu 22.04 LTS operating systems

	Oracle Linux	Astra Linux	Ubuntu 22.04 LTS
Python version	From 3.6 to 3.11. Versions 3.12 and later are not supported.	From 3.6 to 3.11. Versions 3.12 and later are not supported.	From 3.6 to 3.11. Versions 3.12 and later are not supported.
SELinux module	Disabled	Disabled	Disabled

	Oracle Linux	Astra Linux	Ubuntu 22.04 LTS
Package manager	pip3	pip3	pip3

Basic packages	• netaddr
	• firewalld
	• compat-openssl11
	You need to install this package on the Oracle Linux 9 host on which the KUMA Core will be deployed outside the cluster.
	More on upgrading from Oracle Linux 8.x to Oracle Linux 9.x.
	To install the packages, run the following commands:
	`pip3 install netaddr`
	`yum install firewalld`
	`yum install compat-openssl11`
		• python3-apt
		• curl
		• libcurl4
		To install the packages, run the following command:
		`apt install python3-apt curl libcurl4`
			• python3-apt
			• curl
			• libcurl4
			• openssl 1.1.1
			• acl
			To install the packages, run the following command:
			`apt install python3-apt curl libcurl4 acl`
			You can download the openssl1.1.1 package from the official website of Ubuntu and install it using the following command:
			`dpkg -i libssl1.1_1.1.1f-1ubuntu2_amd64.deb`

Dependent packages	No value.	• netaddr • python3-cffi-backend To install the packages, run the following command: `apt install python3-netaddr python3-cffi-backend` If you plan to query Oracle DB databases from KUMA, you need to install the Astra Linux libaio1 package.	• netaddr • python3-cffi-backend To install the packages, run the following command: `apt install python3-netaddr python3-cffi-backend`
Packages that must be installed on a device with the KUMA Core for correct generation and downloading of reports	• nss • gtk2 • atk • libnss3.so • libatk-1.0.so.0 • libxkbcommon • libdrm • at-spi2-atk • mesa-libgbm • alsa-lib • cups-libs • libXcomposite • libXdamage • libXrandr To install the packages, run the following command: `yum install nss gtk2 atk libnss3.so libatk-1.0.so.0 libxkbcommon libdrm at-spi2-atk mesa-libgbm alsa-lib cups-libs libXcomposite libXdamage libXrandr`	• libgtk2.0.0 • libnss3 • libatk-adaptor • libdrm-common • libgbm1 • libxkbcommon0 • libasound2 To install the packages, run the following command: `apt install libgtk2.0.0 libnss3 libatk-adaptor libdrm-common libgbm1 libxkbcommon0 libasound2`	• libatk1.0-0 • libgtk2.0-0 • libatk-bridge2.0-0 • libcups2 • libxcomposite-dev • libxdamage1 • libxrandr2 • libgbm-dev • libxkbcommon-x11-0 • libpangocairo-1.0-0 • libasound2 To install the packages, run the following command: `apt install libatk1.0-0 libgtk2.0-0 libatk-bridge2.0-0 libcups2 libxcomposite-dev libxdamage1 libxrandr2 libgbm-dev libxkbcommon-x11-0 libpangocairo-1.0-0 libasound2`
User permissions required to install the application	No value.	You need to assign the required permissions to the user that will be installing the application: `sudo pdpl-user -i 63 < user that will be installing the application >`	No value.

OSMP Console requirements

OSMP Console Server

For hardware and software requirements, refer to the requirements for a worker node.

Client devices

For a client device, use of OSMP Console requires only a browser.

The minimum screen resolution is 1366x768 pixels.

The hardware and software requirements for the device are identical to the requirements of the browser that is used with OSMP Console.

Browsers:

- Google Chrome 146.0.7680.80 or later (official build)
- Microsoft Edge 146.0.3856.59 or later
- Safari 26.3 on macOS
- "Yandex" Browser 26.3 or later
- Mozilla Firefox Extended Support Release 140.7.0 or later

Network Agent requirements

Minimum hardware requirements:

- CPU with operating frequency of 1 GHz or higher. For a 64-bit operating system, the minimum CPU frequency is 1.4 GHz.
- RAM: 512 MB.
- Available disk space: 1 GB.

Software requirement for Linux-based devices: the Perl language interpreter version 5.10 or higher must be installed.

For inter-process communication in the operating system, the loopback interface must be available.

Network Agent. Supported platforms

Operating systems. Microsoft Windows workstations	Microsoft Windows Embedded POSReady 2009 with latest Service Pack 32-bit Microsoft Windows Embedded 7 Standard with Service Pack 1 32-bit/64-bit Microsoft Windows Embedded 8.1 Industry Pro 32-bit/64-bit Microsoft Windows 10 Enterprise 2015 LTSB 32-bit/64-bit Microsoft Windows 10 Enterprise 2016 LTSB 32-bit/64-bit Microsoft Windows 10 IoT Enterprise 2015 LTSB 32-bit/64-bit Microsoft Windows 10 IoT Enterprise 2016 LTSB 32-bit/64-bit Microsoft Windows 10 Enterprise 2019 LTSC 32-bit/64-bit Microsoft Windows 10 IoT Enterprise version 1703, 1709, 1803, 1809 32-bit/64-bit Microsoft Windows 10 20H2, 21H2 IoT Enterprise 32-bit/64-bit Microsoft Windows 10 IoT Enterprise 32-bit/64-bit Microsoft Windows 10 IoT Enterprise version 1909 32-bit/64-bit Microsoft Windows 10 IoT Enterprise LTSC 2021 32-bit/64-bit Microsoft Windows 10 IoT Enterprise version 1607 32-bit/64-bit Microsoft Windows 10 TH1 (July 2015) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 TH2 (November 2015) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 RS1 (August 2016) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 RS2 (April 2017) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 RS3 (Fall Creators Update, v1709) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 RS4 (April 2018 Update, 17134) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 RS5 (October 2018) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 RS6 (May 2019) Home/Pro/Pro for Workstations/Enterprise/Education 64-bit Microsoft Windows 10 19H1, 19H2 Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 20H1 (May 2020 Update) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 20H2 (October 2020 Update) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 21H1 (May 2021 Update) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 21H2 (October 2021 Update) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 10 22H2 (October 2023 Update) Home/Pro/Pro for Workstations/Enterprise/Education 32-bit/64-bit Microsoft Windows 11 Home/Pro/Pro for Workstations/Enterprise/Education 64-bit Microsoft Windows 11 22H2 Home/Pro/Pro for Workstations/Enterprise/Education 64-bit Microsoft Windows 11 23H2 Home/Pro/Pro for Workstations/Enterprise/Education 64-bit Microsoft Windows 11 24H2 Home/Pro/Pro for Workstations/Enterprise/Education 64-bit Microsoft Windows 11 25H2 Home/Pro/Pro for Workstations/Enterprise/Education 64-bit Microsoft Windows 8.1 Pro/Enterprise 32-bit/64-bit Microsoft Windows 8 Pro/Enterprise 32-bit/64-bit Microsoft Windows 7 Professional/Enterprise/Ultimate/Home Basic/Premium with Service Pack 1 and later 32-bit/64-bit Microsoft Windows XP Professional with Service Pack 2 32-bit/64-bit (supported by Network Agent version 10.5.1781 only) Microsoft Windows XP Professional with Service Pack 3 and later 32-bit (supported by Network Agent version 14.0.0.20023, requires Security Update for Windows XP KB2868626) Microsoft Windows XP Professional for Embedded Systems with Service Pack 3 32-bit (supported by Network Agent version 14.0.0.20023)

Operating systems. Microsoft Windows servers	Microsoft Windows MultiPoint Server 2011 Standard/Premium 64-bit Microsoft Windows Server 2003 SP1 32-bit/64-bit (supported only by Network Agent version 10.5.1781, that you can request through Technical Support) Microsoft Windows Server 2003 SP2 Standard/Enterprise 32-bit/64-bit (supported only by Network Agent version 10.5.1781) Microsoft Windows Server 2003 R2 Standard/Enterprise with SP2 32-bit/64-bit (supported only by Network Agent version 10.5.1781) Microsoft Windows Server 2008 Foundation with Service Pack 2 32-bit/64-bit Microsoft Windows Server 2008 Standard/Enterprise/Datacenter with Service Pack 2 32-bit/64-bit Microsoft Windows Server 2008 R2 Datacenter/Enterprise/Foundation/Standard with Service Pack 1 and later 64-bit Microsoft Windows Server 2012 Server Core/Datacenter/Essentials/Foundation/Standard 64-bit Microsoft Windows Server 2012 R2 Server Core/Datacenter/Essentials/Foundation/Standard 64-bit Microsoft Windows Server 2016 Datacenter/Standard/Server Core (Installation Option) (LTSB) 64-bit Microsoft Windows Server 2019 Standard/Datacenter/Core 64-bit Microsoft Windows Server 2019 RS5 Essentials/Standard 64-bit Microsoft Windows Server 2022 Standard/Datacenter/Core 64-bit Microsoft Windows Server 2025 Standard/Datacenter/Core 64-bit Microsoft Windows Server 2022 21H2 Standard/Datacenter 64-bit Microsoft Windows Storage Server 2019 64-bit Microsoft Windows Small Business Server 2011 Standard 64-bit Microsoft Windows Small Business Server 2011 Essentials 64-bit Microsoft Windows Small Business Server 2011 Premium Add-on 64-bit

Operating systems. Linux	Debian GNU/Linux 10.x (Buster) 32-bit/64-bit Debian GNU/Linux 11.x (Bullseye) 32-bit/64-bit Debian GNU/Linux 12 (Bookworm) 32-bit/64-bit Debian GNU/Linux 13 (Trixie) 64-bit Ubuntu Server 10.04 LTS (Lucid Lynx) 32-bit/64-bit Ubuntu Server 16.04 LTS (Xenial Xerus) 32-bit/64-bit Ubuntu Server 18.04 LTS (Bionic Beaver) 64-bit Ubuntu Server 20.04 LTS (Focal Fossa) 64-bit Ubuntu Server 22.04 LTS (Jammy Jellyfish) 64-bit Ubuntu Server 22.04 LTS ARM 64-bit Ubuntu Server 24.04 LTS (Noble Numbat) 64-bit Ubuntu Desktop 10.04 LTS (Lucid Lynx) 32-bit/64-bit Ubuntu Desktop 16.04 LTS (Xenial Xerus) 32-bit/64-bit CentOS 5.2 32-bit/64-bit CentOS 5.4 32-bit/64-bit CentOS 6.4 32-bit/64-bit CentOS 6.6 32-bit/64-bit CentOS 7.2 and later 64-bit CentOS Stream 8 64-bit CentOS Stream 9 64-bit CentOS Stream 9 ARM 64-bit CentOS Stream 10 64-bit Red Hat Enterprise Linux Server 6.x 32-bit/64-bit Red Hat Enterprise Linux Server 7.2 and later 64-bit Red Hat Enterprise Linux Server 8.x 64-bit Red Hat Enterprise Linux Server 9.x 64-bit Red Hat Enterprise Linux Server 10.x 64-bit SUSE Linux Enterprise Server 12.5 and later (all Service Packs) 64-bit SUSE Linux Enterprise Server 15 (all Service Packs) 64-bit SUSE Linux Enterprise Server 15 (all Service Packs) ARM 64-bit openSUSE Leap 15 64-bit EulerOS 2.0 SP10 64-bit EulerOS 2.0 SP10 ARM 64-bit Astra Linux Special Edition RUSB.10015-01 (operational update 1.4) 32-bit/64-bit Astra Linux Special Edition RUSB.10015-01 (operational update 1.5) 64-bit Astra Linux Special Edition RUSB.10015-01 (operational update 1.6) 64-bit Astra Linux Special Edition RUSB.10015-16 (release 1) (operational update 1.6) 64-bit Astra Linux Special Edition RUSB.10015-17 (operational update 1.7.3) 64-bit Astra Linux Special Edition RUSB.10015-01 (operational update 1.7) 64-bit Astra Linux Special Edition RUSB.10015-01 (operational update 1.8) 64-bit Astra Linux Special Edition RUSB.10015-03 (operational update 7.6) 64-bit Astra Linux Special Edition RUSB.10015-37 (operational update 7.7) 64-bit Astra Linux Special Edition RUSB.10152-02 (operational update 4.7) ARM 64-bit Astra Linux Common Edition (operational update 2.12) 64-bit ALT Workstation 11 64-bit ALT Server 11 64-bit ALT Education 11 64-bit ALT Workstation 10.1 64-bit ALT Server 10.1 64-bit ALT Education 10.1 64-bit ALT SP Server 10 32-bit/64-bit ALT SP Server 10 ARM 64-bit ALT SP Workstation 10 32-bit/64-bit ALT SP Workstation 10 ARM 64-bit ALT 8 SP Server (LKNV.11100-01) 32-bit/64-bit ALT 8 SP Server (LKNV.11100-02) 32-bit/64-bit

	ALT 8 SP Server (LKNV.11100-03) 32-bit/64-bit ALT 8 SP Workstation (LKNV.11100-01) 32-bit/64-bit ALT 8 SP Workstation (LKNV.11100-02) 32-bit/64-bit ALT 8 SP Workstation (LKNV.11100-03) 32-bit/64-bit Oracle Linux 7 64-bit Oracle Linux 8 64-bit Oracle Linux 9 64-bit Linux Mint 21.1 and later 64-bit Linux Mint 22.x 64-bit AlterOS 7.5 and later 64-bit AlterOS 2025 64-bit GosLinux IC6/7.17 64-bit GosLinux IC6/7.2 64-bit SberOS 3.4.0 64-bit SberOS 3.5.0 64-bit Platform V SberLinux OS Server (SLO) 8.10.1 64-bit Platform V SberLinux OS Server (SLO) 9.1 64-bit Platform V SberLinux OS Server (SLO) 9.5.1 64-bit RED OS 7.3 Server 64-bit RED OS 7.3 Certified Edition 64-bit RED OS 8 64-bit RED OS 8 ARM 64-bit ROSA Enterprise Linux Server 7.9 64-bit ROSA Enterprise Linux Desktop 7.9 64-bit ROSA COBALT 7.9 64-bit ROSA CHROME 12 64-bit AlmaLinux 8 and later 64-bit AlmaLinux 9 and later 64-bit Rocky Linux 8 and later 64-bit Rocky Linux 9 and later 64-bit Atlant, Alcyone build, version 2022.02 64-bit MSVSPHERE 9.2 SERVER 64-bit MSVSPHERE 9.2 ARM 64-bit MSVSPHERE 9.4 SERVER 64-bit MSVSPHERE 9.4 ARM 64-bit SynthesisM Server 8.6 64-bit SynthesisM Client 8.6 64-bit OSnova 2.* 64-bit OSnova 3.* 64-bit Kylin 10 64-bit EMIAS 1.0 64-bit Amazon Linux 2 64-bit MosOS 15.5 Arbat 64-bit OS MES (Moscow Electronic School) 12 (for computers and laptops) 64-bit OS MES (Moscow Electronic School) 12 (for interactive panels) 64-bit M OS (Moscow Electronic School) 12 Server 64-bit Mostech 64-bit Mostech Server 64-bit Fedora Linux Server 41 64-bit Fedora Linux Workstation 41 64-bit
Operating systems. macOS	For the list of supported operating systems, refer to the Kaspersky Endpoint Security for Mac Help .

Virtualization platforms	VMware vSphere 6.7.0 VMware vSphere 7.0.3 Citrix XenServer 7.x Citrix XenServer 8.2 Parallels Desktop 18 Oracle VM VirtualBox 7.0.12 Microsoft Hyper-V Server 2019 64-bit Microsoft Hyper-V Server 2022 64-bit KVM (Kernel-based Virtual Machine) Refer to requirements for managed applications for other supported platforms.

On Windows versions older than Microsoft Windows Server 2008 R2 (server versions) and Microsoft Windows 7 (desktop versions) Network Agent does not support the following functionality:

- Updates retranslation (used by distribution points)
- KSN proxy (used by distribution points)
- Vulnerability scan task

On the devices running Microsoft Windows 10 version RS4 or RS5, Kaspersky Security Center might be unable to detect some vulnerabilities in folders where case sensitivity is enabled.

Before installing Network Agent on the devices running Windows 7, Windows Server 2008, Windows Server 2008 R2 or Windows MultiPoint Server 2011, make sure that you have installed the security update KB3063858 for OS Windows (Security Update for Windows 7 (KB3063858), Security Update for Windows 7 for x64-based Systems (KB3063858), Security Update for Windows Server 2008 (KB3063858), Security Update for Windows Server 2008 x64 Edition (KB3063858), Security Update for Windows Server 2008 R2 x64 Edition (KB3063858).

In Microsoft Windows XP, Network Agent might not perform some operations correctly.

You can install or update Network Agent for Windows XP in Microsoft Windows XP only. The supported editions of Microsoft Windows XP and their corresponding versions of the Network Agent are listed in the list of supported operating systems. You can download the required version of the Network Agent for Microsoft Windows XP from this page.

We recommend that you install the same version of the Network Agent for Linux as Open Single Management Platform.

Open Single Management Platform fully supports Network Agent of the same or newer versions.

Network Agent for macOS is provided together with Kaspersky security application for this operating system.

Virtualization configuration

On hypervisor hosts all power-saving optimizations in BIOS/UEFI must be disabled (C-States, P-States, Turbo-Boost, and similar features).

If BIOS/UEFI On hypervisor hosts supports CPU performance settings, they must be configured for maximum performance. Refer to the hardware vendor documentation for guidance.

On hypervisor hosts virtualization support (Intel VTx/VTd, AMD-V/AMD-Vi, etc) must be enabled in BIOS/UEFI.

Transparent Huge Pages (THP) must be enabled.

The I/O scheduler must be set to '`noop`'.

We recommend that you increase the '`max_io_threads`' parameter.

NUMA (Non-Uniform Memory Access)

On hosts with single-socket or low memory, NUMA should be disabled.

On multi-socket systems, NUMA should be enabled (it is advised to add at least 20% extra memory to XDR recommended specifications for NUMA overhead).

VMWare configuration

Enable the '`Reserve all guest memory`' option.

Set minimal values for shared resource parameters (such as CPU and memory shares) to ensure dedicated allocation for each VM.

Use VMXNET3 network adapters for virtual machines to optimize network performance.

Kernel-based Virtual Machine, Proxmox, Nutanix, RED Virtualization, zVirt Node configuration

Configure the hypervisor in host or host-passthrough mode.

Use SCSI drivers for storage subsystems instead of block (blk) drivers.

Do not use shared resources to ensure dedicated resource allocation.

HYPER-V configuration

The vCPU-to-pCore ratio should be configured as 1:1.

Follow the recommendations listed in the articles:

- https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/plan/plan-for-deploying-devices-using-discrete-device-assignment
- https://www.vinchin.com/vm-tips/hyper-v-performance-tuning.html

Compatible applications and solutions

Kaspersky Next XDR Expert can be integrated with the following versions of applications and solutions:

- Kaspersky Security Center Linux 15 or later (as secondary Administration Servers)
- Kaspersky Security Center Windows 14.2 or later (as secondary Administration Servers)
- Kaspersky Anti Targeted Attack Platform 5.1
- Kaspersky Anti Targeted Attack Platform 6.0
- Kaspersky Anti Targeted Attack Platform 7.1
- Kaspersky Anti Targeted Attack Platform 8.0
- Kaspersky Endpoint Security for Windows 12.3 or later (supports file servers)
- Kaspersky Endpoint Security for Windows 12.3 or later
- Kaspersky Endpoint Security for Linux 12.1 or later
- Kaspersky Endpoint Security for Mac 12.0 or later
- Kaspersky CyberTrace 4.2 or later (integration can only be configured in the KUMA Console)
- Kaspersky Endpoint Agent 3.16, 4.0
- Kaspersky Industrial CyberSecurity for Nodes 3.2 or later
- Kaspersky Industrial CyberSecurity for Networks 4.0 (integration can only be configured in the KUMA Console)
- Kaspersky Secure Mail Gateway 2.0 or later (integration can only be configured in the KUMA Console)
- Kaspersky Security for Linux Mail Server 10 or later (integration can only be configured in the KUMA Console)
- Kaspersky Web Traffic Security 6.0 or later (integration can only be configured in the KUMA Console)
- UserGate 7
- Kaspersky Automated Security Awareness Platform
- Kaspersky Threat Intelligence Portal

- Kaspersky NGFW Beta-2 (0.95), 1.0
- Kaspersky Managed Detection and Response

Refer to the Application Support Lifecycle webpage for the versions of the applications.

Known issues

Open Single Management Platform supports management of Kaspersky Endpoint Security for Windows with the following limitations:

- The Adaptive Anomaly Control component is not supported. Open Single Management Platform does not support Adaptive Anomaly Control rules.
- Kaspersky Sandbox components are not supported.

Basic concepts

This section explains basic concepts related to Open Single Management Platform.

Administration Server

Open Single Management Platform components enable remote management of Kaspersky applications installed on client devices.

Devices with the Administration Server component installed will be referred to as *Administration Servers* (also referred to as *Servers*). Administration Servers must be protected, including physical protection, against any unauthorized access.

Administration Server is installed on a device as a service with the following set of attributes:

- With the name `kladminserver_srv`
- Set to start automatically when the operating system starts
- With the `ksc` account or the user account selected during the installation of Administration Server

Refer to the following article for the full list of installation settings: Installing Open Single Management Platform.

Administration Server performs the following functions:

- Storage of the administration groups' structure
- Storage of information about the configuration of client devices
- Organization of repositories for application distribution packages
- Remote installation of applications to client devices and removal of applications

- Updating application databases and software modules of Kaspersky applications
- Management of policies and tasks on client devices
- Storage of information about events that have occurred on client devices
- Generation of reports on the operation of Kaspersky applications
- Deployment of license keys to client devices and storing information about the license keys
- Forwarding notifications about the progress of tasks (such as detection of viruses on a client device)

Naming Administration Servers in the application interface

In the interface of the OSMP Console, Administration Servers can have the following names:

- Name of the Administration Server device, for example: "*device_name*" or "Administration Server: *device_name*".
- IP address of the Administration Server device, for example: "*IP_address*" or "Administration Server: *IP_address*".
- Secondary Administration Servers and virtual Administration Servers have custom names that you specify when you connect a virtual or a secondary Administration Server to the primary Administration Server.
- If you use OSMP Console installed on a Linux device, the application displays the names of the Administration Servers that you specified as trusted in the response file.

You can connect to Administration Server by using OSMP Console.

Hierarchy of Administration Servers

Administration Servers can be arranged in a hierarchy. Each Administration Server can have several secondary Administration Servers (referred to as *secondary Servers*) on different nesting levels of the hierarchy. The root Administration Server can only act as a primary Server. The nesting level for secondary Servers is unrestricted. The administration groups of the primary Administration Server will then include the client devices of all secondary Administration Servers. Thus, isolated and independent sections of networks can be managed by different Administration Servers which are in turn managed by the primary Server.

In a hierarchy, a Linux-based Administration Server can work both as a primary Server and as a secondary Server. The primary Linux-based Server can manage both Linux-based and Windows-based secondary Servers. A primary Windows-based Server can manage a secondary Linux-based Server.

Virtual Administration Servers are a particular case of secondary Administration Servers.

The hierarchy of Administration Servers can be used to do the following:

- Decrease the load on Administration Server (compared to a single installed Administration Server for an entire network).
- Decrease intranet traffic and simplify work with remote offices. You do not have to establish connections between the primary Administration Server and all networked devices, which may be located, for example, in different regions. It is sufficient to install a secondary Administration Server in each network segment, distribute devices among administration groups of secondary Servers, and establish connections between the secondary Servers and the primary Server over fast communication channels.

- Distribute responsibilities among the anti-virus security administrators. All capabilities for centralized management and monitoring of the anti-virus security status in corporate networks remain available.

- Use Open Single Management Platform by service providers. The service provider only needs to install Open Single Management Platform and OSMP Console. To manage a large number of client devices of various organizations, a service provider can add secondary Administration Servers (including virtual Servers) to the hierarchy of Administration Servers.

Each device included in the hierarchy of administration groups can be connected to one Administration Server only. You must independently monitor the connection of devices to Administration Servers. Use the feature for device search in administration groups of different Servers based on network attributes.

Web Server

Open Single Management Platform *Web Server* (hereinafter also referred to as *Web Server*) is a component of Open Single Management Platform that is installed together with Administration Server. Web Server is designed for transmission, over a network, of stand-alone installation packages.

When you create a stand-alone installation package, it is automatically published on Web Server. The link for downloading the stand-alone package is displayed in the list of created stand-alone installation packages. If necessary, you can cancel publication of the stand-alone package or you can publish it on Web Server again. You can send the link to the user in any convenient way, such as by email. By using this link, the user can download the installation package to a local device.

Network Agent

Interaction between Administration Server and devices is performed by the *Network Agent* component of Open Single Management Platform. Network Agent must be installed on all devices on which Open Single Management Platform is used to manage Kaspersky applications.

A device that has Network Agent installed is called a *managed device* or *device*. You can download the installation package for Network Agent from the following sources:

- Administration Server storage (you must have Administration Server installed)

- Kaspersky website

Network Agent for Linux is installed on a device as a service, with the following set of attributes:

- The name "Kaspersky Network Agent"

- Set to start automatically when the operating system starts

- Using the root account

Network Agent for Windows is installed on a device as a service, with the following set of attributes:

- The name "Kaspersky Security Center Network Agent"
- Set to start automatically when the operating system starts
- Using the LocalSystem account

Network Agent for macOS is installed on a device as a service, with the following set of attributes:

- The name "com.kaspersky.klnagent.core"
- Set to start automatically when the operating system starts
- Using the root account

The names of the service processes:

- For Linux:
  - klnagent64.service (for a 64-bit operating system)
  - klnagent.service (for a 32-bit operating system)
- For Windows:
  - klnagent
- For macOS:
  - com.kaspersky.klnagent

By default, Network Agent is installed in the following locations:

- For Linux:
  - 32-bit systems: /opt/kaspersky/klnagent/
  - 64-bit systems: /opt/kaspersky/klnagent64/
- For Windows:
  - 32-bit systems: C:\Program Files\Kaspersky Lab\NetworkAgent
  - 64-bit systems: C:\Program Files (x86)\Kaspersky Lab\NetworkAgent
- For macOS:
  - 64-bit systems: /Library/Application Support/Kaspersky Lab/klnagent/Binaries/klnagent

For Windows devices, you can specify a different folder for the installation of Network Agent in the settings of the installation package. However, for Linux devices, Network Agent can only be installed in the default directory.

The Network Agent installation folder also contains utilities for managing and diagnosing the operation of Network Agent, such as the klmover and klnagchk utilities.

When you install Administration Server, the server version of Network Agent is automatically installed together with Administration Server. Nevertheless, to manage the Administration Server device as any other managed device, install Network Agent for Linux on the Administration Server device. In this case, Network Agent for Linux is installed and works independently from the server version of Network Agent that you installed together with Administration Server.

Network Agent synchronizes the managed device with the Administration Server. We recommend that you set the synchronization interval (also referred to as the *heartbeat*) to 15 minutes per 10,000 managed devices.

Administration groups

An *administration group* (hereinafter also referred to as *group*) is a logical set of managed devices combined on the basis of a specific trait for the purpose of managing the grouped devices as a single unit within Open Single Management Platform.

All managed devices within an administration group are configured to do the following:

- Use the same application settings (which you can specify in group policies).
- Use a common operating mode for all applications through the creation of group tasks with specified settings. Examples of group tasks include creating and installing a common installation package, updating the application databases and modules, scanning the device on demand, and enabling real-time protection.

A managed device can belong to only one administration group.

You can create hierarchies that have any degree of nesting for Administration Servers and groups. A single hierarchy level can include secondary and virtual Administration Servers, groups, and managed devices. You can move devices from one group to another without physically moving them. For example, if a worker's position in the enterprise changes from that of accountant to developer, you can move this worker's device from the Accountants administration group to the Developers administration group. Thereafter, the device will automatically receive the application settings required for developers.

Managed device

A *managed device* is a device running Linux, Windows, or macOS and which has Network Agent installed. You can manage such devices by creating tasks and policies for applications installed on these devices. You can also receive reports from managed devices.

You can make a managed device function as a distribution point and as a connection gateway.

A device can be managed by only one Administration Server. The number of devices that can be managed by one Administration Server depends on configuration of the device that hosts the Administration Server and on DBMS restrictions.

Unassigned device

An *unassigned device* is a device on the network that has not been included in any administration group. You can perform some actions on unassigned devices, for example, move them to administration groups or install applications on them.

When a new device is discovered on your network, this device goes to the **Unassigned devices** administration group. You can configure rules for devices to be moved automatically to other administration groups after the devices are discovered.

Administrator's workstation

Devices on which OSMP Console Server is installed are referred to as *administrator's workstations*. Administrators can use these devices for centralized remote management of Kaspersky applications installed on client devices.

There are no restrictions on the number of administrator's workstations. From any administrator's workstation, you can manage administration groups of several Administration Servers on the network at once. You can connect an administrator's workstation to an Administration Server (physical or virtual) of any level of the hierarchy.

You can include an administrator's workstation in an administration group as a client device.

Within the administration groups of any Administration Server, the same device can function as an Administration Server client, an Administration Server, or an administrator's workstation.

Management web plug-in

A special component—the *management web plug-in*—is used for remote administration of Kaspersky software by means of OSMP Console. Hereinafter, a management web plug-in is also referred to as a *management plug-in*. A management plug-in is an interface between OSMP Console and a specific Kaspersky application. With a management plug-in, you can configure tasks and policies for the application.

The management plug-in provides the following:

- Interface for creating and editing application tasks and settings
- Interface for creating and editing policies and policy profiles for remote and centralized configuration of Kaspersky applications and devices
- Transmission of events generated by the application
- OSMP Console functions for displaying operational data and events of the application, and statistics relayed from client devices

Policies

A *policy* is a set of Kaspersky application settings that are applied to an administration group and its subgroups. You can install several Kaspersky applications on the devices of an administration group. Open Single Management Platform provides a single policy for each Kaspersky application in an administration group. A policy has one of the following statuses:

The status of the policy

Status	Description
Active	The current policy that is applied to the device. Only one policy may be active for a Kaspersky application in each administration group. Devices apply the settings values of an active policy for a Kaspersky application.
Inactive	A policy that is not currently applied to a device.
Out-of-office	If this option is selected, the policy becomes active when the device leaves the corporate network.

Policies function according to the following rules:

- Multiple policies with different values can be configured for a single application.
- Only one policy can be active for the current application.
- A policy can have child policies.

Generally, you can use policies as preparations for emergency situations, such as a virus attack. For example, if there is an attack via flash drives, you can activate a policy that blocks access to flash drives. In this case, the current active policy automatically becomes inactive.

In order to prevent maintaining multiple policies, for example, when different occasions assume changing of several settings only, you may use policy profiles.

A *policy profile* is a named subset of policy settings values that replaces the settings values of a policy. A policy profile affects the effective settings formation on a managed device. *Effective settings* are a set of policy settings, policy profile settings, and local application settings that are currently applied for the device.

Policy profiles function according to the following rules:

- A policy profile takes effect when a specific activation condition occurs.
- Policy profiles contain values of settings that differ from the policy settings.
- Activation of a policy profile changes the effective settings of the managed device.
- A policy can include a maximum of 100 policy profiles.

Policy profiles

Sometimes it may be necessary to create several instances of a single policy for different administration groups; you might also want to modify the settings of those policies centrally. These instances might differ by only one or two settings. For example, all the accountants in an enterprise work under the same policy—but senior accountants are allowed to use flash drives, while junior accountants are not. In this case, applying policies to devices only through the hierarchy of administration groups can be inconvenient.

To help you avoid creating several instances of a single policy, Open Single Management Platform enables you to create *policy profiles*. Policy profiles are necessary if you want devices within a single administration group to run under different policy settings.

A policy profile is a named subset of policy settings. This subset is distributed on target devices together with the policy, supplementing it under a specific condition called the *profile activation condition*. Profiles only contain settings that differ from the "basic" policy, which is active on the managed device. Activation of a profile modifies the settings of the "basic" policy that were initially active on the device. The modified settings take values that have been specified in the profile.

Tasks

Open Single Management Platform manages Kaspersky security applications installed on devices by creating and running *tasks*. Tasks are required for installing, launching, and stopping applications, scanning files, updating databases and software modules, and performing other actions on applications.

Tasks for a specific application can be created only if the management plug-in for that application is installed.

Tasks can be performed on the Administration Server and on devices.

The following tasks are performed on the Administration Server:

- Automatic distribution of reports
- Downloading of updates to the repository of the Administration Server
- Backup of Administration Server data
- Maintenance of the database

The following types of tasks are performed on devices:

- *Local tasks*—Tasks that are performed on a specific device

  Local tasks can be modified either by the administrator, by using OSMP Console, or by the user of a remote device (for example, through the security application interface). If a local task has been modified simultaneously by the administrator and the user of a managed device, the changes made by the administrator will take effect because they have a higher priority.

- *Group tasks*—Tasks that are performed on all devices of a specific group

  Unless otherwise specified in the task properties, a group task also affects all subgroups of the selected group. A group task also affects (optionally) devices that have been connected to secondary and virtual Administration Servers deployed in the group or any of its subgroups.

- *Global tasks*—Tasks that are performed on a set of devices, regardless of whether they are included in any group

For each application, you can create any number of group tasks, global tasks, or local tasks.

You can make changes to the settings of tasks, view the progress of tasks, and copy, export, import, and delete tasks.

A task is started on a device only if the application for which the task was created is running.

Results of tasks are saved in the Syslog event log and the Open Single Management Platform event log, both centrally on the Administration Server and locally on each device.

Do not include private data in task settings. For example, avoid specifying the domain administrator password.

Task scope

The *scope of a task* is the set of devices on which the task is performed. The types of scope are as follows:

- For a *local task*, the scope is the device itself.
- For an *Administration Server task*, the scope is the Administration Server.
- For a *group task*, the scope is the list of devices included in the group.

When creating a *global task*, you can use the following methods to specify its scope:

- Specifying certain devices manually.

  You can use an IP address (or IP range) or DNS name as the device address. You can also select networked devices detected by Administration Server.

- Importing a list of devices from a .txt file with the device addresses to be added (each address must be placed on an individual line).

  If you import a list of devices from a file or create a list manually, and if devices are identified by their names, the list can only contain devices for which information has already been entered into the Administration Server database. Moreover, the information must have been entered when those devices were connected or during device discovery.

- Specifying a device selection.

  Over time, the scope of a task changes as the set of devices included in the selection change. A selection of devices can be made on the basis of device attributes, including software installed on a device, and on the basis of tags assigned to devices. Device selection is the most flexible way to specify the scope of a task.

  Tasks for device selections are always run on a schedule by the Administration Server. These tasks cannot be run on devices that lack connection to the Administration Server. Tasks whose scope is specified by using other methods are run directly on devices and therefore do not depend on the device connection to the Administration Server.

  Tasks for device selections are not run on the local time of a device; instead, they are run on the local time of the Administration Server. Tasks whose scope is specified by using other methods are run on the local time of a device.

How local application settings relate to policies

You can use policies to set identical values of the application settings for all devices in a group.

The values of the settings that a policy specifies can be redefined for individual devices in a group by using local application settings. You can set only the values of settings that the policy allows to be modified, that is, the unlocked settings.

The value of a setting that the application uses on a client device is defined by the lock position (🔒) for that setting in the policy:

- If a setting modification is locked, the same value (defined in the policy) is used on all client devices.
- If a setting modification is unlocked, the application uses a local setting value on each client device instead of the value specified in the policy. The setting can then be changed in the local application settings.

This means that, when a task is run on a client device, the application applies settings that have been defined in two different ways:

- By task settings and local application settings, if the setting is not locked against changes in the policy.
- By the group policy, if the setting is locked against changes.

Local application settings are changed after the policy is first applied in accordance with the policy settings.

Distribution point

Distribution point (previously known as update agent) is a device with Network Agent installed that is used for update distribution, remote installation of applications, and retrieval of information about networked devices.

> The features and use cases of Network Agent installed on a device used as a distribution point vary depending on the operating system.

A distribution point can perform the following functions:

- Distribute updates and installation packages received from the Administration Server to client devices within the group (including distribution through multicasting using UDP). Updates can be received either from the Administration Server or from Kaspersky update servers. In the latter case, an update task must be created for the distribution point.

  Distribution points accelerate update distribution and free up Administration Server resources.

- Distribute policies and group tasks through multicasting using UDP.

- Act as a gateway for connection to the Administration Server for devices in an administration group.

  If a direct connection between managed devices within the group and the Administration Server cannot be established, you can use the distribution point as a connection gateway to the Administration Server for this group. In this case, managed devices connect to the connection gateway, which in turn connects to the Administration Server.

The presence of a distribution point that functions as connection gateway does not block the option of a direct connection between managed devices and the Administration Server. If the connection gateway is not available, but direct connection with the Administration Server is technically possible, managed devices are connected to the Administration Server directly.

- Poll the network to detect new devices and update information about existing ones. A distribution point can apply the same device discovery methods as the Administration Server.

- Perform remote installation of applications by Kaspersky and other software vendors, including installation on client devices without Network Agent.

  This feature allows you to remotely transfer Network Agent installation packages to client devices located on networks to which the Administration Server has no direct access.

- Act as a proxy server participating in Kaspersky Security Network (KSN).

  You can enable KSN proxy server on distribution point side to make the device act as a KSN proxy server. In this case, the KSN proxy service is run on the device.

Files are transmitted from the Administration Server to a distribution point over HTTP or, if SSL connection is enabled, over HTTPS. Using HTTP or HTTPS results in a higher level of performance, compared to SOAP, through cutting traffic.

Devices with Network Agent installed can be assigned distribution points either manually (by the administrator), or automatically (by the Administration Server). The full list of distribution points for specified administration groups is displayed in the report about the list of distribution points.

The scope of a distribution point is the administration group to which it has been assigned by the administrator, as well as its subgroups of all levels of embedding. If multiple distribution points have been assigned in the hierarchy of administration groups, Network Agent on the managed device connects to the nearest distribution point in the hierarchy.

If distribution points are assigned automatically by the Administration Server, it assigns them by broadcast domains, not by administration groups. This occurs when all broadcast domains are known. Network Agent exchanges messages with other Network Agents in the same subnet and then sends Administration Server information about itself and other Network Agents. Administration Server can use that information to group Network Agents by broadcast domains. Broadcast domains are known to Administration Server after more than 70% Network Agents in administration groups are polled. Administration Server polls broadcast domains every two hours. After distribution points are assigned by broadcast domains, they cannot be re-assigned by administration groups.

If the administrator manually assigns distribution points, they can be assigned to administration groups or network locations.

Network Agents with an active connection profile do not participate in broadcast domain detection.

Open Single Management Platform assigns each Network Agent a unique IP multicast address that differs from every other address. This allows you to avoid network overload that might occur due to IP overlaps. IP multicast addresses that were assigned in previous versions of the application will not be changed.

If two or more distribution points are assigned to a single network area or to a single administration group, one of them becomes the active distribution point, and the rest become standby distribution points. The active distribution point downloads updates and installation packages directly from the Administration Server, while standby distribution points receive updates from the active distribution point only. In this case, files are downloaded once from the Administration Server and then are distributed among distribution points. If the active distribution point becomes unavailable for any reason, one of the standby distribution points becomes active. The Administration Server automatically assigns a distribution point to act as standby.

The distribution point status (*Active*/*Standby*) is displayed with a check box in the klnagchk report.

A distribution point requires at least 4 GB of free disk space. If the free disk space of the distribution point is less than 2 GB, Open Single Management Platform creates a security issue with the *Warning* importance level. The security issue will be published in the device properties, in the **Security issues** section.

Running remote installation tasks on a device assigned as a distribution point requires additional free disk space. The volume of free disk space must exceed the total size of all installation packages to be installed.

Running any updating (patching) tasks and vulnerability fix tasks on a device assigned as a distribution point requires additional free disk space. The volume of free disk space must be at least twice the total size of all patches to be installed.

Devices functioning as distribution points must be protected, including physical protection, against any unauthorized access.

Connection gateway

A *connection gateway* is a Network Agent acting in a special mode. A connection gateway accepts connections from other Network Agents and tunnels them to the Administration Server through its own connection with the Server. Unlike an ordinary Network Agent, a connection gateway waits for connections from the Administration Server rather than establishes connections to the Administration Server.

A connection gateway can receive connections from up to 10,000 devices.

You have two options for using connection gateways:

- We recommend that you install a connection gateway in a demilitarized zone (DMZ). For other Network Agents installed on out-of-office devices, you need to specially configure a connection to Administration Server through the connection gateway.

  A connection gateway does not in any way modify or process data that is transmitted from Network Agents to Administration Server. Moreover, it does not write this data into any buffer and therefore cannot accept data from a Network Agent and later forward it to Administration Server. If Network Agent attempts to connect to Administration Server through the connection gateway, but the connection gateway cannot connect to Administration Server, Network Agent perceives this as if Administration Server is inaccessible. All data remains on Network Agent (not on the connection gateway).

  A connection gateway cannot connect to Administration Server through another connection gateway. It means that Network Agent cannot simultaneously be a connection gateway and use a connection gateway to connect to Administration Server.

All connection gateways are included in the list of distribution points in the Administration Server properties.

- You can also use connection gateways within the network. For example, automatically assigned distribution points also become connection gateways in their own scope. However, within an internal network, connection gateways do not provide considerable benefit. They reduce the number of network connections received by Administration Server, but do not reduce the volume of incoming data. Even without connection gateways, all devices could still connect to Administration Server.

Core

The *Core* is the central component of KUMA that serves as the foundation upon which all other services and components are built. The Core's console provides a graphical user interface that is intended for everyday use by operators/analysts and for configuring the entire system.

The Core allows you to:

- create and configure services, or components, of the program, as well as integrate the necessary software into the system;
- manage program services and user accounts in a centralized way;
- visualize statistical data on the program;
- investigate security threats based on the received events.

Storage

A KUMA *storage* is used to store normalized events so that they can be quickly and continually accessed from KUMA for the purpose of extracting analytical data. Access speed and continuity are ensured through the use of the ClickHouse technology. This means that a *storage* is a ClickHouse cluster bound to a KUMA storage service. ClickHouse clusters can be supplemented with cold storage disks.

When choosing a ClickHouse cluster configuration, consider the specific event storage requirements of your organization. For more information, please refer to the ClickHouse documentation.

In repositories, you can create *spaces*. The spaces enable to create a data structure in the cluster and, for example, store the events of a certain type together.

Collector

A *collector* is an application component that receives messages from event sources, processes them, and transmits them to a storage, correlator, and/or third-party services to identify alerts.

For each collector, you need to configure one connector and one normalizer. You can also configure an unlimited number of additional Normalizers, Filters, Enrichment rules, and Aggregation rules. To enable the collector to send normalized events to other services, specific destinations must be added. Normally, two destinations are used: the storage and the correlator.

The collector operation algorithm includes the following steps:

1. **Receiving messages from event sources**

   To receive messages, you must configure an active or passive connector. The passive connector can only receive messages from the event source, while the active connector can initiate a connection to the event source, such as a database management system.

   Connectors can also vary by type. The choice of connector type depends on the transport protocol for transmitting messages. For example, for an event source that transmits messages over TCP, you must install a TCP type connector.

   The following connector types are available:

   - internal
   - tcp
   - udp
   - netflow
   - sflow
   - nats-jetstream
   - kafka
   - http
   - sql
   - file
   - diode
   - ftp
   - nfs
   - wmi
   - wec
   - snmp
   - elastic
   - etw

2. **Event parsing and normalization**

   Events received by the connector are processed using the normalizer and normalization rules set by the user. The choice of normalizer depends on the format of the messages received from the event source. For example, you must select a CEF-type root normalizer for a source that sends events in CEF format.

The following normalizers are available:

- JSON
- CEF
- Regexp
- Syslog (as per RFC3164 and RFC5424)
- CSV
- Key-value
- XML
- NetFlow v5
- NetFlow v9
- IPFIX (v10)
- SQL

3. **Filtering of normalized events**

   You can configure filters that allow you to select the events that meet the specified conditions for further processing.

4. **Enrichment and conversion of normalized events**

   Enrichment rules let you to supplement event contents with information from internal and external sources.

   The following enrichment sources are available:

   - constants
   - cybertrace
   - dictionaries
   - dns
   - events
   - ldap
   - templates
   - timezone data
   - geographic data

   Mutation rules let you convert event field contents in accordance with the defined criteria.

The following conversion methods are available:

- lower—converts all characters to lower case.
- upper—converts all characters to upper case.
- regexp—extracts a substring using RE2 regular expressions.
- substring—gets a substring based on the specified numbers of the start and end positions.
- replace—replaces text with the entered string.
- trim—deletes the specified characters.
- append—adds characters to the end of the field value.
- prepend—adds characters to the beginning of the field value.

5. **Aggregation of normalized events**

   You can configure aggregation rules to reduce the number of similar events that are transmitted to the storage and/or the correlator. Configuring aggregation rules lets you combine several events into one event. This helps you reduce the load on the services responsible for further event processing, conserves storage space and the license quota for events per second (EPS). For example, you can aggregate into one event all events involving network connections made using the same protocol (transport and application layers) between two IP addresses and received during a specified time interval.

6. **Transmission of normalized events**

   After all the processing stages are completed, the event is sent to configured destinations.

Correlator

The *Correlator* is a program component that analyzes normalized events. Information from active lists and/or dictionaries can be used in the correlation process.

The data obtained by analysis is used to carry out the following tasks:

- Active lists content management
- Sending correlation events to configured destinations

Event correlation is performed in real time.

The operating principle of the correlator is based on an event signature analysis. This means that every event is processed according to the correlation rules set by the user. When the program detects a sequence of events that satisfies the conditions of the correlation rule, it creates a correlation event and sends it to the Storage. The correlation event can also be sent to the correlator for reanalysis, which allows you to customize the correlation rules so that they are triggered by the results of a previous analysis. Results of one correlation rule can be used by other correlation rules.

You can distribute correlation rules and the active lists they use among correlators, thereby sharing the load between services. In this case, the collectors will send normalized events to all available correlators.

The correlator operation algorithm consists of the following steps:

1. **Obtaining an event**

   The correlator receives a normalized event from the collector or from another service.

2. **Applying correlation rules**

   You can configure correlation rules so they are triggered based on a single event or a sequence of events. If no alert was detected using the correlation rules, the event processing ends.

3. **Responding to an alert**

   You can specify actions that the program must perform when an alert is detected. The following actions are available in the program:

   - Event enrichment
   - Operations with active lists
   - Sending notifications
   - Storing correlation event

4. **Sending a correlation event**

   When the program detects a sequence of events that satisfies the conditions of the correlation rule, it creates a correlation event and sends it to the storage. Event processing by the correlator is now finished.

Events

Events are information security events registered on the monitored elements of the organization's IT infrastructure. For example, events include login attempts, interactions with a database, and sensor information broadcasts. Each separate event may seem meaningless, but when considered together they form a bigger picture of network activities to help identify security threats. This is the core functionality of KUMA.

KUMA receives events from logs and restructures their information, making the data from different event sources consistent (this process is called normalization). Afterwards, the events are filtered, aggregated, and later sent to the correlator service for analysis and to the Storage for retaining. When KUMA recognizes specific event or a sequences of events, it creates *correlation events*, that are also analyzed and retained. If an event or sequence of events indicates a potential security threat, KUMA creates an alert. This alert consists of a warning about the threat and all related data that should be investigated by a security officer.

Throughout their life cycle, events undergo conversions and may receive different names. Below is a description of a typical event life cycle:

The first steps are carried out in a collector.

1. Raw event. The original message received by KUMA from an event source using a Connector is called a *raw event*. This is an unprocessed message and it cannot be used yet by KUMA. To fit into the KUMA pipeline, raw events must be normalized into the KUMA data model. That's what the next stage is for.

2. Normalized event. A normalizer transforms 'raw' event data in accordance with the KUMA data model. After this conversion, the original message becomes a *normalized event* and can be used by KUMA for analysis. From here on, only normalized events are used in KUMA. Raw events are no longer used, but they can be kept as a part of normalized events inside the `Raw` field.

The program has the following normalizers:

- JSON
- CEF
- Regexp
- Syslog (as per RFC3164 and RFC5424)
- CSV/TSV
- Key-value
- XML
- Netflow v5, v9, IPFIX (v10), sFlow v5
- SQL

At this point normalized events can already be used for analysis.

3. Event destination. After the Collector service have processed an event, it is ready to be used by other KUMA services and sent to the KUMA Correlator and/or Storage.

The next steps of the event life cycle are completed in the correlator.

Event types:

1. Base event. An event that was normalized.
2. Aggregated event. When dealing with a large number of similar events, you can "merge" them into a single event to save processing time and resources. They act as base events, but In addition to all the parameters of the parent events (events that are "merged"), aggregated events have a counter that shows the number of parent events it represents. Aggregated events also store the time when the first and last parent events were received.
3. Correlation event. When a sequence of events is detected that satisfies the conditions of a correlation rule, the program creates a *correlation event*. These events can be filtered, enriched, and aggregated. They can also be sent for storage or looped into the Correlator pipeline.
4. Audit event. Audit events are created when certain security-related actions are completed in KUMA. These events are used to ensure system integrity. They are automatically placed in a separate storage space and stored for at least 365 days.
5. Monitoring event. These events are used to track changes in the amount of data received by KUMA.

Alerts

In KUMA, an alert is created when a sequence of events is received that triggers a correlation rule. Correlation rules are created by KUMA analysts to check incoming events for possible security threats, so when a correlation rule is triggered, it's a warning there may be some malicious activity happening. Security officers should investigate these alerts and respond if necessary.

KUMA automatically assigns the severity to each alert. This parameter shows how important or numerous the processes are that triggered the correlation rule. Alerts with higher severity should be dealt with first. The severity value is automatically updated when new correlation events are received, but a security officer can also set it manually. In this case, the alert severity is no longer automatically updated.

Alerts have related events linked to them, making alerts enriched with data from these events. KUMA also offers drill down functionality for alert investigations.

You can create incidents based on alerts.

Alert management in KUMA is described in this section.

Incidents

If the nature of the data received by KUMA or the generated correlation events and alerts indicate a possible attack or vulnerability, the symptoms of such an event can be combined into an *incident*. This allows security experts to analyze threat manifestations in a comprehensive manner and facilitates response.

You can assign a category, type, and severity to incidents, and assign incidents to data protection officers for processing.

Incidents can be exported to NCIRCC and MDR.

Resources

Resources are KUMA components that contain parameters for implementing various functions: for example, establishing a connection with a given web address or converting data according to certain rules. Like parts of an erector set, these components are assembled into resource sets for services that are then used as the basis for creating KUMA services.

Services

Services are the main components of KUMA that work with events: receiving, processing, analyzing, and storing them. Each service consists of two parts that work together:

- One part of the service is created inside the KUMA Console based on the set of resources for services.
- The second part of the service is installed in the network infrastructure where the KUMA system is deployed as one of its components. The server part of a service can consist of multiple instances: for example, services of the same agent or storage can be installed on multiple devices at once.

Parts of services are connected to each other via the service ID.

KUMA agents

KUMA *agents* are services that are used to forward raw events from servers and workstations to KUMA destinations.

Types of agents:

- wmi agents are used to receive data from remote Windows devices using Windows Management Instrumentation. They are installed to Windows assets.
- wec agents are used to receive Windows logs from a local device using Windows Event Collector. They are installed to Windows assets.
- tcp agents are used to receive data over the TCP protocol. They are installed to Linux and Windows assets.
- udp agents are used to receive data over the UDP protocol. They are installed to Linux and Windows assets.
- nats agents are used for NATS communications. They are installed to Linux and Windows assets.
- kafka agents are used for Kafka communications. They are installed to Linux and Windows assets.
- http agents are used for communication over the HTTP protocol. They are installed to Linux and Windows assets.
- file agents are used to get data from a file. They are installed to Linux assets.
- ftp agents are used to receive data over the File Transfer Protocol. They are installed to Linux and Windows assets.
- nfs agents are used to receive data over the Network File System protocol. They are installed to Linux and Windows assets.
- snmp agents are used to receive data over the Simple Network Management Protocol. They are installed to Linux and Windows assets.
- diode agents are used together with data diodes to receive events from isolated network segments. They are installed to Linux and Windows assets.
- etw agents are used to receive Event Tracing for Windows data. They are installed to Windows assets.

About severity

Severity reflects the relative importance of security-sensitive activity detected by a KUMA correlator. It shows the order in which multiple alerts should be processed, and indicates whether senior security officers should be involved.

The Correlator automatically assigns severity to correlation events and alerts based on correlation rule settings. The severity of an alert also depends on the assets related to the processed events because correlation rules take into account the severity of a related asset's category. If the alert or correlation event does not have linked assets with a defined severity or does not have any related assets at all, the severity of this alert or correlation event is equal to the severity of the correlation rule that triggered them. The alert or the correlation event severity is never lower than the severity of the correlation rule that triggered them.

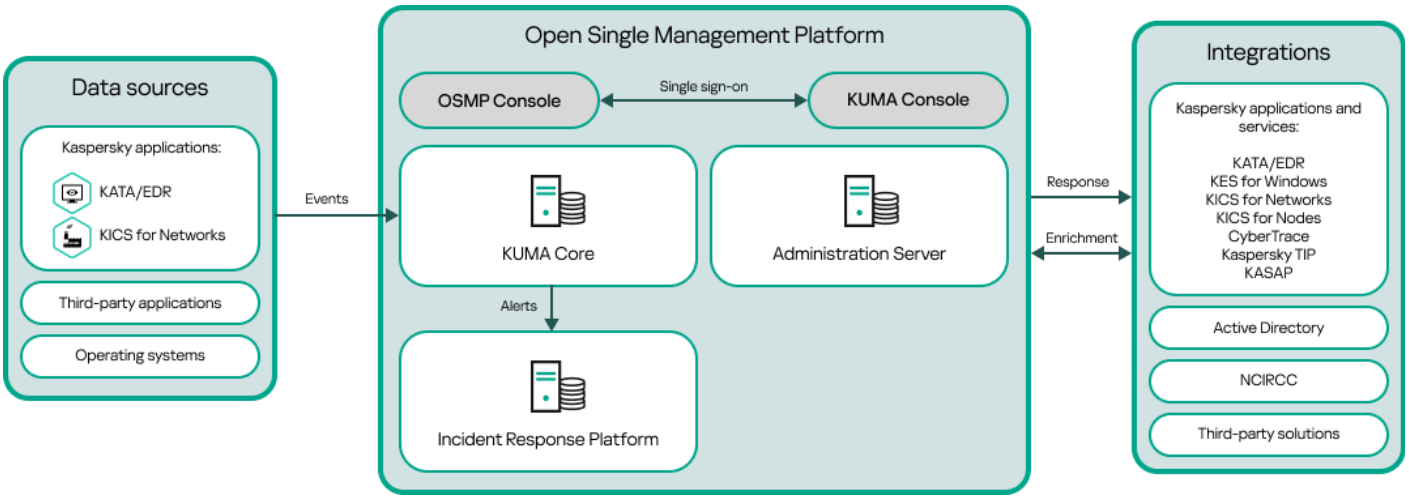
Alert severity can be changed manually. The severity of alerts changed manually is no longer automatically updated by correlation rules.

Possible severity values:

- Low
- Medium
- High
- Critical

Architecture of Open Single Management Platform

This section provides a description of the components of Open Single Management Platform and their interaction.



Open Single Management Platform architecture

Open Single Management Platform comprises the following main components:

- **Open Single Management Platform** (OSMP). The technology basis on which Kaspersky Next XDR Expert is built. OSMP integrates all of the solution components and provides interaction between the components. OSMP is scalable and supports integration with both Kaspersky applications and third-party solutions.

- **OSMP Console**. Provides a web interface for OSMP.

- KUMA Core. The central component of KUMA. KUMA receives, processes, and stores information security events and then analyzes the events by using correlation rules. As a result of the analysis, if the conditions of a correlation rule are met, KUMA creates an alert and sends it to Incident Response Platform.

- **Incident Response Platform**. An Open Single Management Platform component that allows you to create incidents automatically or manually, manage alert and incident life cycle, assign alerts and incidents to SOC analysts, and respond to the incidents automatically or manually, including responses through playbooks.

- Administration Server (also referred to as *Server*). The key component of endpoint protection of a client organization. Administration Server provides centralized deployment and management of endpoint protection through EPP-applications, and allows you to monitor the endpoint protection status.

- **Data sources**. Information security hardware and software that generates the events. After you integrate Kaspersky Next XDR Expert with the required data sources, KUMA receives the events to store and analyze them.

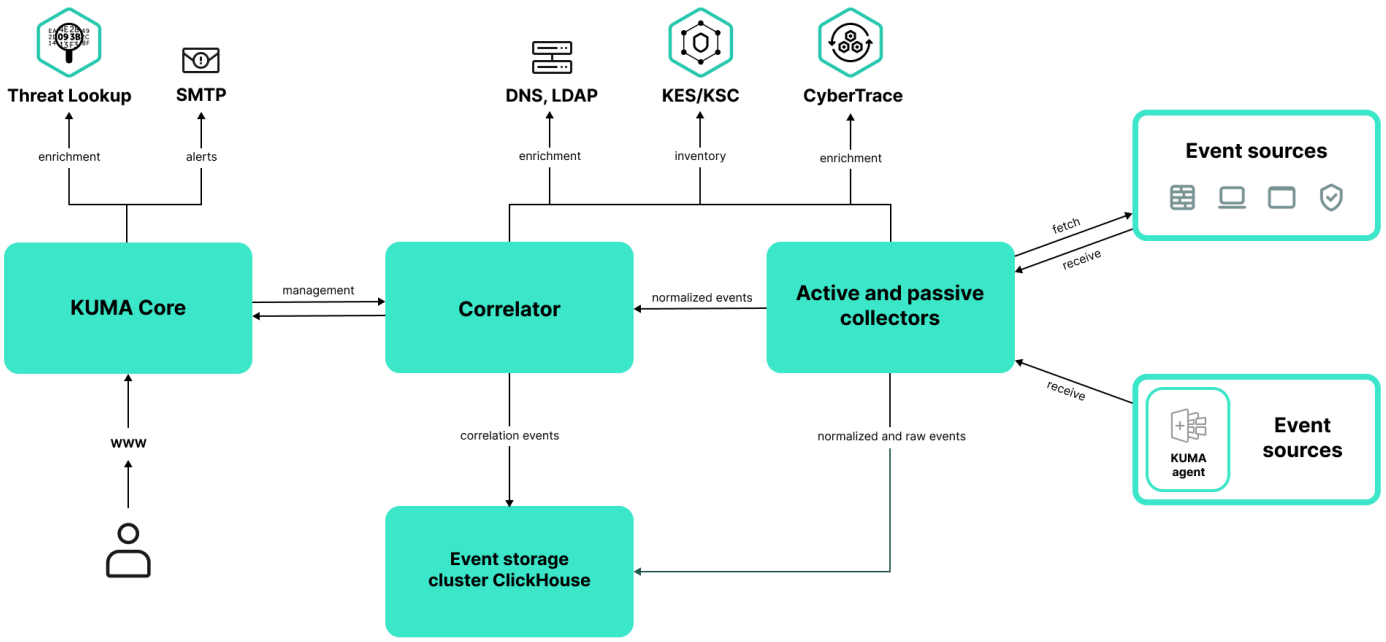
- Integrations. Kaspersky applications and third-party solutions integrated with OSMP. Through integrated solutions, an SOC analyst can enrich the data required for incident investigation, and then respond to incidents.

SIEM component architecture

The standard program installation includes the following components:

- The *Core* that includes a graphical interface to monitor and manage the settings of system components.
- One or more *Collectors* that receive messages from event sources and parse, normalize, and, if required, filter and/or aggregate them.
- A *Correlator* that analyzes normalized events received from Collectors, performs the necessary actions with active lists, and creates alerts in accordance with the correlation rules.
- The *Storage*, which contains normalized events and registered incidents.

Events are transmitted between components over optionally encrypted, reliable transport protocols. You can configure load balancing to distribute load between service instances, and it is possible to enable automatic switching to the backup component if the primary one is unavailable. If all components are unavailable, events are saved to the hard disk buffer and sent later. The size of the buffer in the file system for temporary storage of events can be changed.



KUMA architecture

OSMP Console interface

Kaspersky Next XDR Expert is managed through the OSMP Console interface.

The OSMP Console window contains the following items:

- Main menu in the left part of the window
- Upper pane in the upper part of the window
- Work area in the right part of the window

Main menu

The main menu contains the following sections:

- **Administration Server**. Displays the name of the Administration Server that you are currently connected to. You can open the Administration Server properties by clicking the Administration Server name.
- **Console navigation**. Displays the main menu map. By default, **Console navigation** is set as a home page. You can change these setting.
- **Monitoring**. Provides an overview of your infrastructure, protection statuses, and statistics, including threat hunting, alerts and incidents, and playbooks. Also, this section allows you to add and configure KUMA resources and services.
- **Asset management**. Contains tools for assets, as well as tasks and Kaspersky application policies.
- **Users & roles**. Allows you to manage users and roles, configure user rights by assigning roles to the users, and associate policy profiles with roles.
- **Operations**. Contains a variety of operations, including application licensing, viewing and managing encrypted drives and encryption events, and third-party application management. This also provides you access to application repositories.
- **Discovery & deployment**. Allows you to poll the network to discover client devices, and distribute the devices to administration groups manually or automatically. This section also contains the Protection deployment wizard.
- **Marketplace**. Contains information about the entire range of Kaspersky business solutions and allows you to select the ones you need, and then proceed to purchase those solutions at the Kaspersky website.
- **Settings**. Contains settings to integrate Kaspersky Next XDR Expert with other Kaspersky applications and allows you to create API tokens. It also contains settings related to displaying interface elements depending on features being used, as well as to interface language. Also, from this section you can open the KUMA settings window that contains parameters for KUMA configuration, including KUMA integration settings, license key management, and access options.

If necessary, you can minimize the main menu to icons by clicking the chevron sign (<<) at the bottom of the screen.

Upper pane

The upper pane is always displayed in all console sections and cannot be hidden.

The left side of the pane contains the application name and logo. If you click the application name or logo, the OSMP Console home page opens.

On the right side of the pane, the following settings are displayed:

- Notification icon (). If you click the icon, a pane with notifications opens.
- The menu of your account. Contains a link to Kaspersky Next XDR Expert Help. It also allows you to sign out of Kaspersky Next XDR Expert, view your work status, the OSMP Console version and the list of installed management web plug-ins.

Work area

The work area displays the information you choose to view in the sections of the OSMP Console interface window. It also contains control elements that you can use to configure how the information is displayed.

Changing the language of the OSMP Console interface

You can select the language of the OSMP Console interface.

To change the interface language:

1. In the main menu, go to **Settings** → **Language**.
2. Select one of the supported localization languages.

Changing the OSMP Console home page

The home page is displayed after you sign in to OSMP Console.

By default, the **Console navigation** section is set as the home page. If necessary, you can change the home page.

To change the home page:

1. In **Console navigation** section, hover the mouse cursor over the item that you want to set as the home page.
2. After the home () icon is displayed, click the icon.

   The home page is changed, an appropriate message is displayed on the screen, and the item selected as a home page is marked with the home () icon.

The **Marketplace** and **Settings** sections cannot be set as a home page.

Adding and removing bookmarks

You can add frequently used sections of OSMP Console to bookmarks to access them quickly from the **Bookmarks** section in the main menu.

> If you did not add any section to bookmarks, the **Bookmarks** section is not displayed in the main menu.

You can add to bookmarks only the sections that display pages. For example, if you go to **Asset management** → **Assets (Devices)**, a page with the table of devices opens, which means you can add this section to bookmarks. If a window or no element is displayed after you select the section in the main menu, then you cannot add such a section to bookmarks.

To add a section to bookmarks:

1. In the main menu, hover the mouse cursor over the section you want to pin.
2. After the bookmark ( ) icon is displayed, click the icon.

   The section is displayed in the **Bookmarks** section.

To remove a section from bookmarks:

1. In the main menu, go to the **Bookmarks** section.
2. Hover the mouse cursor over the section you want to remove, and then click the bookmark ( ) icon.

   The section is removed from bookmarks.

You can also add and remove sections from bookmarks in the **Console navigation** section by hovering the mouse cursor over them and clicking the bookmark ( ) icon. The selected item will be marked with the bookmark ( ) icon and displayed in the **Bookmarks** section in the main menu. If you want to remove the item from the bookmarks, click the bookmark ( ) icon.

> The **Bookmarks** section is displayed only in the main menu. It is never displayed on the **Console navigation** page.

Licensing

This section covers the main aspects of Open Single Management Platform licensing.

About the End User License Agreement

The End User License Agreement (License Agreement) is a binding agreement between you and AO Kaspersky Lab stipulating the terms on which you may use the application.

Carefully read the License Agreement before you start using the application.

You can view the terms of the End User License Agreement by using the following methods:

- During installation of Open Single Management Platform.
- By reading the license.txt document. This document is included in the application distribution kit.

You accept the terms of the End User License Agreement by confirming that you agree with the End User License Agreement when installing the application. If you do not accept the terms of the License Agreement, cancel application installation and do not use the application.

About the license key

A *license key* is a sequence of bits that you can apply to activate and then use the application in accordance with the terms of the End User License Agreement. License keys are generated by Kaspersky specialists.

You can add a license key to the application using one of the following methods: by applying a *key file* or by entering an *activation code*. The license key is displayed in the application interface as a unique alphanumeric sequence after you add it to the application.

The license key may be blocked by Kaspersky in case the terms of the License Agreement have been violated. If the license key has been blocked, you need to add another one if you want to use the application.

A license key may be active or additional (or reserve).

An *active license key* is a license key that is currently used by the application. An active license key can be added for a trial or commercial license. The application cannot have more than one active license key.

An *additional (or reserve) license key* is a license key that entitles the user to use the application, but is not currently in use. The additional license key automatically becomes active when the license associated with the current active license key expires. An additional license key can be added only if an active license key has already been added.

A license key for a trial license can be added as an active license key. A license key for a trial license cannot be added as an additional license key.

About the activation code

An *activation code* is a unique sequence of 20 letters and numbers. You have to enter an activation code in order to add a license key for activating Open Single Management Platform. You receive the activation code at the email address that you provided when you bought Open Single Management Platform or requested the trial version of Open Single Management Platform.

To activate the application by using the activation code, you need internet access in order to connect to Kaspersky activation servers.

If you have lost your activation code after installing the application, contact the Kaspersky partner from whom you purchased the license.

About the key file

A *key file* is a file with the .key extension provided to you by Kaspersky. Key files are designed to activate the application by adding a license key.

You receive a key file at the email address that you provided when you bought Open Single Management Platform or ordered the trial version of Open Single Management Platform.

You do not need to connect to Kaspersky activation servers in order to activate the application with a key file.

You can restore a key file if it has been accidentally deleted. You may need a key file to register a Kaspersky CompanyAccount, for example.

To restore your key file, perform any of the following actions:

- Contact the license seller.
- Receive a key file through Kaspersky website by using your available activation code.

License limits

When you purchase a Kaspersky Next XDR Expert license, you determine the number of users you want to protect. You can exceed the license limit by no more than 5%. If you exceed the license limit by more than 5%, the extra devices and extra accounts are added to the **Restricted assets** list.

If the license limit is exceeded, a notification appears at the top of the OSMP Console.

It is not possible to launch response actions or playbooks for restricted assets.

To view the list of restricted assets:

1. In the main menu, go to **Settings** → **Tenants**.
2. In the **Tenants** section, click the Root tenant.

   The Root tenant's properties window opens.
3. Select the **Licenses** tab.
4. Click the link with the number of restricted assets.

The **Restricted assets** window opens.

> The list shows a maximum of 2000 restricted assets (the first 1000 devices and the first 1000 accounts).

Kaspersky Next XDR Expert license features

The Kaspersky Next XDR Expert license extends the capabilities of a Kaspersky EDR Expert license, adding advanced tools for automated response and incident management.

Additional features included in a Kaspersky Next XDR Expert license:

- **Advanced responses**—automated responses to alerts and incidents through custom scripts or UserGate.
- **XDR Responses**—an extended catalog of XDR response actions.
- **Advanced case management**—Service Level Agreement (SLA), Configuring the automatic assignment of alerts and incidents, and cross-tenant child-incidents support.

The Kaspersky Next XDR Expert license provides more flexible and scalable incident management, enabling integration into complex infrastructures and advanced automation of response processes.

Activating Kaspersky Next XDR Expert

After you install Kaspersky Next XDR Expert, you must activate the application in the Administration Server properties.

To activate Kaspersky Next XDR Expert:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.
2. On the **General** tab, select the **License keys** section.
3. Under **Current license**, click the **Select** button.
4. In the window that opens, select the license key that you want to use to activate Kaspersky Next XDR Expert. If the license key is not listed, click the **Add new license key** button, and then specify a new license key.

5. If necessary, you can also add a reserve license key. To do this, under **Reserve license key**, click the **Select** button, and then select an existing license key or add a new one. Note that you cannot add a reserve license key if there is no active license key.

6. Click the **Save** button.

Adding a license key for the SIEM component

You can add an application license key in the Kaspersky Security Center Web Console. If you use a key file to activate the application, two license keys, may be present at the same time: an active key and a reserve key. If you use an activation code, there is only one license key. In KUMA, a group of an active and a reserve license keys or one activation code is called a *license group*. Accordingly, you can add a license group using two key files or using an activation code.

If multiple license groups are added to an instance of KUMA, the active license keys of all license groups are automatically merged. This creates a *merged license*. License keys of such a license are merged in accordance with the following rules:

- The **EPS available** values of all active license keys are added together. The resulting EPS value is specified in the merged license and becomes available to all tenants.

  If any license key included in the merged license expires, the EPS limit value of that license key is automatically subtracted from the total EPS limit of the merged license. Accordingly, when a new active license key is added to the merged license, the EPS limit value of that license key is automatically added to the total EPS limit.

- The set of modules available in tenant licenses must be the same. If a license contains a module that is not available in the other licenses, this module is not available in the merged license. Accordingly, such a module is not available to any of the tenants.

  The same rule applies when adding new license keys to the merged license. If a module is available in the new license key, but is not available in the merged license, this module does not become available under the merged license. If a module that is available in the merged license is not available in the new license key, this module will become unavailable under the merged license. When adding a license key with a mismatched set of modules, KUMA displays a warning and adds such a license key only after your confirmation.

- A merged license can include up to 50 active keys.

Only users with the General Administrator role can add a license key.

Adding a new license group

To add a new license group:

1. In the main application window, go to the **Settings** → **KUMA** → **Open KUMA settings** section.

   This opens the KUMA settings window.

2. Under **Other**, click **License**.

3. Click **Add license key**.

4. This opens a window; in that window, enter a name for the new license group.

5. Click **Add**.

6. Click the name of the added license group.

7. In the window that is opened, select how you want to add a license key:

   - Using an activation code.

     1. Enter the activation code.

     2. If your company uses a proxy server to connect to the internet, enter proxy server information. An internet connection is required to activate the application.

   - Using a key file.

     1. In the **Active license key** section, click the **Upload** button, then select the key file (file with the KEY extension).

     2. In the **Reserve license key** section, add the reserve license key in the same way.

8. Click **Add**.

The license group is added to the list, and the application is activated.

Adding a reserve license key

To add a reserve license key to a license group:

1. In the main application window, go to the **Settings** → **KUMA** → **Open KUMA settings** section.

   This opens the KUMA settings window.

2. Under **Other**, click **License**.

3. Click the name of the license group to which you want to add the reserve license key.

4. This opens a window; in that window, in the **Reserve license key** section, click the **Upload** button, then select the key file (file with the KEY extension).

5. Click **Add**.

Replacing the active license key

If you want to add an active license key instead of the key that has already been added, delete the current license key first. Then add the new license key.

Viewing information about license keys in use

To view active and reserve license keys:

1. In the main menu, go to **Settings** → **Tenants**.

2. In the **Tenants** section, click the root tenant.

   The root tenant's properties window opens.

3. Select the **Licenses** tab.

   The active and reserve license keys are displayed.

The displayed license key is applied to all child tenants of the root tenant. Specifying a separate license key for a child tenant is not available. The properties window for child tenants does not include the **Licenses** tab.

If the license keys limit is exceeded, a notification is shown, and the information about the license key shows a warning.

You can click the **Go to Administration Server** button to manage Kaspersky Next XDR Expert license keys.

On the **Licenses** tab, you can also view the list of licensed objects. To do this, click the ⓘ button.

The availability of the licensed object depends on the purchased license type.

Renewing licenses for Kaspersky applications

You can renew licenses for Kaspersky Next XDR Expert and included Kaspersky applications, such as Kaspersky Unified Monitoring and Analysis Platform. You can renew licenses that have expired or are going to expire within 30 days.

An email with an archive containing the new license keys will be sent to your email address after you purchase a new Kaspersky Next XDR Expert license.

To renew a license of Kaspersky Next XDR Expert:

1. Extract the new license keys from the archive sent to your email address.

2. Follow the steps described in Activating Kaspersky Next XDR Expert.

   The license is renewed.

> If you need to renew the licenses of the included Kaspersky applications, you must add new license keys to the web interfaces of these solutions.

For how to renew a license of Kaspersky Unified Monitoring and Analysis Platform, see the Adding a license key to the program web interface section of the Kaspersky Unified Monitoring and Analysis Platform Help.

In OSMP Console, the notifications are displayed when a license is about to expire, according to the following schedule:

- 30 days before the expiration
- 7 days before the expiration
- 3 days before the expiration
- 24 hours before the expiration
- When a license has expired

Removing a license key of the SIEM component

In KUMA, you can remove an added license key from the Kaspersky Security Center Web Console (for example, if you need to replace the current license key with a different key). After the license key is removed, collectors immediately stop receiving and processing events. Old events remain available. Creating and modifying resources, creating and running services also becomes impossible. This functionality will be re-activated after you add a license key.

> Only users with the General Administrator role can delete license keys.

To delete an added license key:

1. In the main application window, go to the **Settings** → **KUMA** → **Open KUMA settings** section.

   This opens the KUMA settings window.

2. Under **Other**, click **License**.

3. Select the check box for the license key that you want to delete and click the trash can icon ( ).

4. This opens a confirmation window; in that window, click **Delete**.

   The license key will be removed from the application.

About data provision

Data processed locally

Kaspersky Next XDR Expert is designed to optimize threat detection, incident investigation, threat response (including automatic), and proactive threat hunting in real time.

Kaspersky Next XDR Expert performs the following main functions:

- Receiving, processing, and storing information security events.
- Analysis and correlation of incoming data.
- Incidents and alerts investigation, manual response.
- Automatic response by using the predefined and custom playbooks.
- Event-based threat hunting in real time.

To perform its main functions, Kaspersky Next XDR Expert can receive, store and process the following information:

- Information about the devices on which all Kaspersky Next XDR Expert components are installed:
  - Technical specifications: device name, MAC address, operating system vendor, operating system build number, OS kernel version, required installed packages, account rights, service management tool type, and port status. This data is obtained by Kaspersky Deployment Toolkit during installation.
  - Technical specifications: IPv4 address. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.
  - Device access data: account names and SSH keys for accessing devices where Kaspersky Next XDR Expert components are installed. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.
  - Kaspersky Next XDR Expert account name and password. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.
  - Database access data: IP/DNS name, port, user name, and password. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.
  - KUMA inventory file and license keys. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.
  - DNS zone. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.
  - Certificates for secure connection of devices to OSMP components. This data is specified by the user in the Kaspersky Deployment Toolkit configuration file.

  Information is saved in the installation log, which is stored in the Kaspersky Deployment Toolkit database. The installation log of the initial infrastructure is saved to a file on the user's device. The storage period is indefinite; the installation log file will be deleted when Kaspersky Next XDR Expert is uninstalled. User names and passwords are stored in an encrypted form.

- Information about incident status models, including the predefined model.
- Information about incident types: incident type name, description and other general information.
- Information about user accounts: full name and email address. The user enters data in the OSMP Console. The data is stored in the database until the user deletes it.
- Integration token data.
- Information about tenants: tenant name, parent tenant name, description. The user enters data in the OSMP Console. The data is stored in the database until the user deletes it.
- Information about certificates: client certificates for applications acting as an Endpoint Agent, OSMP server certificate, KUMA collector server certificate.
- Information on devices in the organization's network.
  - OSMP receives the data when the relevant integration is configured. You can add assets in the following ways:
    - Import assets:
      - On demand from MaxPatrol.
      - On a schedule from Kaspersky Security Center Core and Kaspersky Industrial CyberSecurity for Networks.
    - Create assets manually through the OSMP Console or via the API.
  - OSMP stores the following information about devices:
    - Technical characteristics (general properties, OS, hardware details, and installed Kaspersky applications).
    - Detected vulnerabilities and device protection status.
    - Source-specific data.
    - Device categories and their criticality levels.
    - AI score and status.

- Alerts and incidents data:
  - Alert data: triggered rules, compliance with the MITRE matrix, alert status, resolution, assigned operator or group of operators, affected assets (devices and accounts), observables (IP, MD5, SHA256, URL, DNS domain, or DNS name) user name, host name, comments, changelog, files. This information is generated in the OSMP Console automatically, based on correlation events obtained from Kaspersky Unified Monitoring and Analysis Platform.
    - Additional data is also stored depending on the alert type:
      - For alerts generated based on file scan results: names and hash values of scanned files; Anti-Malware and Sandbox decisions; information about local file actions and network activity triggered by the file during Sandbox analysis; a password-protected archive containing the scanned file object.
      - For alerts generated by TAA (IOA) rules by Kaspersky: information about triggered IOA rules, IOA rule descriptions, response recommendations, and possible causes of false positives (rule information is not provided by the user and is distributed by Kaspersky).
      - For alerts generated by IOC detection technology: names and severity of triggered IOC rules; objects detected during IOC device scanning, including IP addresses, DNS records, registry keys and values, routing table entries, process and file information, service details, and user name and ID.
      - For alerts generated by YARA detection technology: names and severity of triggered YARA rules; objects detected during YARA scanning of the device, including file and process information.
  - Incident data: linked alerts, child incidents, triggered rules, compliance with the MITRE matrix, incident status, resolution, assigned operator or group of operators, affected assets (devices and accounts), observables (from the alert), comments, changelog, files, incident type, metrics for incident sorting, responding, resolution, and details of the triggered IOA rule, including its name, description, response recommendations, and possible causes of false positives. This information is generated in the OSMP Console automatically, according to the rules or manually by the user.
  - Data on configuring the segmentation rules for generating incidents from alerts: the name and the rule triggering conditions, the template for the name of a new incident, a rule description, and the rule launch priority. The user enters data in the OSMP Console.
  - Information about notification templates: template name, message subject, message template, template description, detection rules, the selected detection technologies to notify the user about, and the severity of alerts and incidents. When the detection rules are triggered, notifications are sent. The user enters data in the OSMP Console.
- Expiration date settings for alerts and incidents.
- Incident investigation graph and process tree. OSMP stores the following user-defined information about the process trees built for events: the names of the created process trees and comments.
- Playbook data:
  - Playbook operational data, including data on response action parameters: name, description, tags, trigger, and algorithm. The user enters data in the OSMP Console.
  - Data on the execution of response actions within a playbook: data from integrated systems, data from devices.
  - The full response history of alerts and incidents, including response actions not associated with an alert or incident.

The data listed above is stored in the database for three days and then deleted. Data is completely deleted when Kaspersky Next XDR Expert is uninstalled.

- Credentials for accessing shared resources. The credentials are stored in an encrypted form and are distributed to devices to support response actions, such as obtaining memory dumps and disk images.

- Information about files obtained from devices through response actions or uploaded manually by users, including file name or full path, size, hash, tenant, source device or uploading user, description, scanning status (AV, YARA, Sandbox), and scan results.

- Information about prevention rules applied to devices. This includes the rule name, hash, tenant, the devices to which the rule applies, the rule status (enabled or disabled), and the option to notify the endpoint user when a file matching the prevention rule is executed.

- Information about YARA rules used for scanning on devices. This includes the rule name, tenant, description, and the file scanning option (enabled or disabled), which determines whether the rule is applied during server-side scanning of files obtained from devices or uploaded by users. The rules are stored as files in the OSMP object storage and are delivered to endpoints when the YARA scan response action is executed.

- Information about IOC rules used for scanning on devices. This includes the rule name, severity, tenant, and the scheduled scanning option (enabled or disabled). The rules are stored as files in the OSMP object storage and are delivered to endpoints when the IOC scan response action is executed. For scheduled scanning, the scan schedule is also delivered to endpoints, including the enable and disable option, scan start time, and maximum scan duration.

- File object scanning settings, including a catalog of commonly used client passwords for archive decryption and the update source for Asset Management and Sandbox databases. The settings are configured by the user in the OSMP Console.

- When KPSN is used in OSMP, the following data is processed: the name or IP address of the server where the KPSN API is deployed, as well as the certificate used to connect to the API.

- Integration settings data (both with Kaspersky solutions or services, and with third-party solutions that participate in Kaspersky Next XDR Expert scenarios):

  - Kaspersky Threat Intelligence Portal integration: API access token for connecting to Kaspersky Threat Intelligence Portal, cache retention period, whether the connection is through a proxy, or service type. The user enters data in the OSMP Console.

  - KATA and KEDR integration: KATA and KEDR server address: IP address or host name, port, unique ID for connecting to KATA and KEDR, certificate file, and a private key for connecting to KATA and KEDR. The user enters data in the OSMP Console.

  - Connection to the host where the custom script will be run: IP address or host name, port, user name and SSH key, and password or key. The user enters data in the OSMP Console.

  - OSMP Administration Server integration: Administration Server name, full path to the Administration Server in the hierarchy. The user enters data in the OSMP Console.

  - Kaspersky CyberTrace integration: IPv4 address or hostname and port through which Kaspersky CyberTrace is available, name, and password. The user enters data in the OSMP Console.

  - Kaspersky Automated Security Awareness Platform (KASAP) integration: API access token for connecting to KASAP, KASAP portal URL, KASAP administrator email address, and whether the connection is through a proxy. The user enters data in the OSMP Console.

- Active Directory integration: addresses of domain controllers, user name and password for connecting to domain controllers, and certificate. The user enters data in the OSMP Console.
- External system integration (such as UserGate): account name and SSH key or password for remote access to the client device.
- Kaspersky Sandbox integration: reference configurations of virtual machine operating systems on which the Sandbox is deployed, the Sandbox server IP address, the Sandbox server certificate, the server name displayed in the OSMP Console, and Sandbox rules for file analysis. The user enters the data in the OSMP Console.

- IP address from which the user connects to the OSMP Console. This data is logged automatically in the OSMP Console and is stored until the expiration of revisions of user-edited objects.
- Information about the OSMP installation hierarchy configuration.

For detailed information about other data received, stored, and processed to perform the main functions of Kaspersky Next XDR Expert, refer to the application Help:

- Kaspersky Security Center 16.1 Linux
- Kaspersky Unified Monitoring and Analysis Platform
- Kaspersky EDR Expert

The above data is stored in the database until the user deletes it. Data related to an alert or incident is stored for the entire retention period of the alert or incident. All other data is stored indefinitely until the user deletes it or deletes the tenant to which the data is associated. This data is completely deleted when Kaspersky Next XDR Expert is uninstalled.

All data processed locally can be transferred to Kaspersky only through the dump files, trace files, or log files of Kaspersky Next XDR Expert components, including log files created by installers and utilities. The dump files, trace files, or log files of Kaspersky Next XDR Expert components contain personal or confidential data. The dump files, trace files, and log files are stored on the devices in an unencrypted form. The dump files, trace files, or log files are not transferred to Kaspersky automatically, but an administrator may transfer those files to Kaspersky manually by request from Technical Support to resolve issues related to Kaspersky Next XDR Expert performance. Kaspersky protects any information received in accordance with the law and applicable Kaspersky rules. Data is transmitted over a secure channel. The default storage term for this information (rotation period) is 7 days.

Data transferred to AO Kaspersky Lab

By following the links from the OSMP Console to Kaspersky Next XDR Expert Help, the user agrees to the automatic transfer of the following data to Kaspersky:

- Kaspersky Next XDR Expert code
- Kaspersky Next XDR Expert version
- Kaspersky Next XDR Expert localization

To assign a training course to an employee, Kaspersky Next XDR Expert transfers the following data to Kaspersky Automated Security Awareness Platform:

- user email
- Kaspersky Automated Security Awareness Platform ID
- training group ID

To obtain additional alert data, Kaspersky Next XDR Expert transfers the type and value of observables related to alerts, incidents and events to Kaspersky Threat Intelligence Portal.

To obtain information about the assessment of a detected threat when following a link from an alert, incident, or event card, Kaspersky Next XDR Expert transfers the following data to the Kaspersky Threats service: decision name.

To receive incident detection and response services, Kaspersky Next XDR Expert transfers the following data to the Kaspersky Managed Detection and Response service:

- When integration with Kaspersky Managed Detection and Response is enabled, OSMP temporarily stores the following data until it is successfully transmitted, and then transfers it to the Kaspersky Managed Detection and Response infrastructure: the device identifier from which the rule-triggered event was received, the user identifier, and the Kaspersky Managed Detection and Response customer identifier.
- When using the Kaspersky Managed Detection and Response solution from OSMP, the following additional information from alerts is transferred to the Kaspersky Managed Detection and Response infrastructure: the names and hash values of scanned files; Anti-Malware and Sandbox detection results; information about local file actions and network activity generated by the file during Sandbox analysis; a password-protected archive containing the scanned file object; and events that triggered IOA rules, which may include file and application names on devices, registry key names and values, and user names.
- When using the Kaspersky Managed Detection and Response solution from OSMP, the following incident-related information is transferred to the Kaspersky Managed Detection and Response infrastructure at the user's request: the incident name, the date and time of creation and last update, information about triggered rules and associated MITRE tactics and techniques, the name of the operator assigned to the incident, the names and identifiers of affected devices, and events included in alerts. These events may include file names on customer devices, registry key names, application names, and user names.

  All of this data is transferred to Kaspersky Managed Detection and Response only after the user accepts the Kaspersky Managed Detection and Response Data Processing Agreement (DPA), which is confirmed when Kaspersky Managed Detection and Response is activated in the Kaspersky Managed Detection and Response interface.

To enhance information protection capabilities, Kaspersky Next XDR Expert transfers KSN statistical data generated during OSMP operation to the Kaspersky Security Network service. The scope of transferred data is specified in the KSN Agreement, which is included in the application distribution package. Participation in KSN is optional and controlled by the OSMP user. Before transferring, KSN statistical data is temporarily cached on servers hosting OSMP components.

When using Kaspersky Private Security Network, KSN statistical data is cached on servers hosting OSMP components in the same scope as when using Kaspersky Security Network. If the Kaspersky Private Security Network server is located outside the organization, the cached KSN statistical data may be transferred outside the organization. The responsibility for ensuring the security of this data lies with the Kaspersky Private Security Network administrator.

Data transferred to third parties

By following the link from the alert or incident details for receiving information about the MITRE tactics or technique, the following information about MITRE tactics or techniques is transferred to the MITRE website: ID and type.

To perform response actions through external systems (for example, UserGate) by executing third-party scripts on remote client devices, Kaspersky Next XDR Expert transfers the following information to external systems: the types and values of observed objects from alerts and incidents.

To check the reputation of files and URLs when following a link from an event card, Kaspersky Next XDR Expert transfers the following information to the VirusTotal service: the file hashes (FileMd5 and FileSha256); the remote host name (NetUrlHostname); the destination URI for HTTP requests (NetUri); the destination IPv4 or IPv6 address (NetDstIp); the URL requested in a DNS query (DnsResourceName); and the requested DNS server (DnsServerIp).

Data provision in Open Single Management Platform

Data processed locally

Open Single Management Platform is designed for centralized execution of basic administration and maintenance tasks on an organization's network. Open Single Management Platform provides the administrator with access to detailed information about the organization's network security level; Open Single Management Platform lets an administrator configure all the components of protection based on Kaspersky applications. Open Single Management Platform performs the following main functions:

- Detecting devices and their users on the organization's network
- Creating a hierarchy of administration groups for device management
- Installing Kaspersky applications on devices
- Managing the settings and tasks of installed applications
- Activating Kaspersky applications on devices
- Managing user accounts
- Viewing information about the operation of Kaspersky applications on devices
- Viewing reports

To perform its main functions Open Single Management Platform can receive, store, and process the following information:

- Information about the devices on the organization's network received through scanning of Active Directory or Samba domain controllers or through scanning of IP intervals. Administration Server gets data independently or receives data from Network Agent.
- Information from Active Directory and Samba about organizational units, domains, users, and groups. Administration Server gets data by itself or receives data from Network Agent assigned to work as a distribution point.

- Details of managed devices. Network Agent transfers the data listed below from the device to Administration Server. The user enters the display name and description of the device in the OSMP Console interface:
  - Technical specifications of the managed device and its components required for device identification: device display name and description, Windows domain name and type (for devices belonging to a Windows domain), device name in Windows environment (for devices belonging to a Windows domain), DNS domain and DNS name, IPv4 address, IPv6 address, network location, MAC address, operating system type, whether the device is a virtual machine together with hypervisor type, and whether the device is a dynamic virtual machine as part of VDI.
  - Other specifications of managed devices and their components required for audit of managed devices: operating system architecture, operating system vendor, operating system build number, operating system release ID, operating system location folder, if the device is a virtual machine—the virtual machine type, name of the virtual Administration Server that manages the device.
  - Details of actions on managed devices: date and time of the last update, time the device was last visible on the network, restart waiting status, and time the device was turned on.
  - Details of device user accounts and their work sessions.
- Data received by running remote diagnostics on a managed device: trace files, system information, details of Kaspersky applications installed on the device, dump files, event logs, the results of running the diagnostic scripts received from Kaspersky Technical Support.
- Distribution point operation statistics if the device is a distribution point. Network Agent transfers data from the device to Administration Server.
- Distribution point settings entered by the User in OSMP Console.
- Details of Kaspersky applications installed on the device. The managed application transfers data from the device to Administration Server through Network Agent:
  - Settings of Kaspersky applications installed on the managed device: Kaspersky application name and version, status, real-time protection status, last device scan date and time, number of threats detected, number of objects that failed to be disinfected, availability and status of the application components, details of Kaspersky application settings and tasks, information about the active and reserve license keys, application installation date and ID.
  - Application operation statistics: events related to the changes in the status of Kaspersky application components on the managed device and to the performance of tasks initiated by the application components.
  - Device status defined by the Kaspersky application.
  - Tags assigned by the Kaspersky application.
- Data contained in events from Open Single Management Platform components and Kaspersky managed applications. Network Agent transfers data from the device to Administration Server.
- Settings of Open Single Management Platform components and Kaspersky managed applications presented in policies and policy profiles. The User enters data in the OSMP Console interface.
- Task settings of Open Single Management Platform components and Kaspersky managed applications. The User enters data in the OSMP Console interface.

- Data processed by the System management feature. Network Agent transfers from the device to Administration Server the following information:
  - Information about the hardware detected on managed devices (Hardware registry).

    If Network Agent is installed on a device running Windows, it sends to the Administration Server the following information about the device hardware:

    - RAM
    - Mass storage devices
    - Motherboard
    - CPU
    - Network adapters
    - Monitors
    - Video adapter
    - Sound card

    If Network Agent is installed on a device running Linux, it sends to the Administration Server the following information about the device hardware, if this information is provided by the operating system:

    - Total RAM volume
    - Total volume of mass storage devices
    - Motherboard
    - CPU
    - Network adapters
  - Information about the software installed on managed devices (Software registry). The software can be compared with the information about the executable files detected on the devices by the Application Control function.
- User categories of applications. The User enters data in the OSMP Console interface.
- Details of executable files detected on managed devices by the Application Control feature. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.
- Information about encrypted Windows-based devices and the encryption status. The managed application transfers data from the device to Administration Server through Network Agent.
- Details of data encryption errors on Windows-based devices performed using the Data encryption feature of Kaspersky applications. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.
- Details of files placed in Backup. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.

- Details of files placed in Quarantine. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.

- Details of files requested by Kaspersky specialists for detailed analysis. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.

- Details of external devices (memory units, information transfer tools, information hardcopy tools, and connection buses) installed or connected to the managed device and detected by the Device Control feature. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.

- Information about encrypted devices and the encryption status. A managed application transfers data from the device to Administration Server through Network Agent.

- Information about data encryption errors on the devices. The encryption is performed by the Encryption data function of Kaspersky applications. A managed application transfers data from the device to Administration Server through Network Agent. The full list of data is provided in the Online Help of the corresponding application.

- List of managed programmable logic controllers (PLCs). The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.

- Data required for creation of a threat development chain. The managed application transfers data from the device to Administration Server through Network Agent. A complete list of data is provided in the Help files of the corresponding application.

- Details of the entered activation codes and key files. The User enters data in the Administration Console or OSMP Console interface.

- User accounts: name, description, full name, email address, main phone number, and password. The User enters data in the OSMP Console interface.

- Revision history of management objects. The User enters data in the OSMP Console interface.

- Registry of deleted management objects. The User enters data in the OSMP Console interface.

- Installation packages created from the file, as well as installation settings. The User enters data in the OSMP Console interface.

- Data required for the display of announcements from Kaspersky in OSMP Console. The User enters data in the OSMP Console interface.

- Data required for the functioning of plug-ins of managed applications in OSMP Console and saved by the plug-ins in the Administration Server database during their routine operation. The description and ways of providing the data are provided in the Help files of the corresponding application.

- OSMP Console user settings: localization language and theme of the interface, Monitoring panel display settings, information about the status of notifications (Already read / Not yet read), status of columns in spreadsheets (Show / Hide), Training mode progress. The User enters data in the OSMP Console interface.

- Certificate for secure connection of managed devices to the Open Single Management Platform components. The User enters data in the OSMP Console interface.

- Information on which Kaspersky legal agreement terms have been accepted by the user.

- The Administration Server data that the User enters in the OSMP Console or program interface Kaspersky Security Center OpenAPI.

- Any data that the User enters in the OSMP Console interface.

The data listed above can be present in Open Single Management Platform if one of the following methods is applied:

- The User enters data in the OSMP Console interface.

- Network Agent automatically receives data from the device and transfers it to Administration Server.

- Network Agent receives data retrieved by the Kaspersky managed application and transfers it to Administration Server. The lists of data processed by Kaspersky managed applications are provided in the Help files for the corresponding applications.

- Administration Server gets the information about the networked devices by itself or receives data from Network Agent assigned to work as a distribution point.

The listed data is stored in the Administration Server database. User names and passwords are stored in encrypted form.

All data processed locally can be transferred to Kaspersky only through dump files, trace files, or log files of Open Single Management Platform components, including log files created by installers and utilities.

The dump files, trace files, or log files of Open Single Management Platform components contain arbitrary data of Administration Server, Network Agent, and OSMP Console. The files may contain personal or confidential data. The dump files, trace files, or log files are stored on the devices in an unencrypted form. The dump files, trace files, or log files are not transferred to Kaspersky automatically, but an administrator may transfer those files to Kaspersky manually by request from Technical Support to resolve issues related to Open Single Management Platform performance.

Kaspersky protects any information received in accordance with law and applicable Kaspersky rules. Data is transmitted over a secure channel.

Following the links in the Administration Console or OSMP Console, the User agrees to the automatic transfer of the following data:

- Open Single Management Platform code

- Open Single Management Platform version

- Open Single Management Platform localization

- License ID

- License type

- Whether the license was purchased through a partner

The list of data provided via each link depends on the purpose and location of the link.

Kaspersky uses the received data in anonymized form and for general statistics only. Summary statistics are generated automatically from the originally received information and do not contain any personal or confidential data. As soon as new data is accumulated, the previous data is wiped (once a year). Summary statistics are stored indefinitely.

Data provision in Kaspersky Unified Monitoring and Analysis Platform

Data provided to third parties

KUMA functionality does not involve automatic provision of user data to third parties.

Locally processed data

Kaspersky Unified Monitoring and Analysis Platform (hereinafter KUMA or "program") is an integrated software solution that includes the following primary functions:

- Receiving, processing, and storing information security events.
- Analysis and correlation of incoming data.
- Search within the obtained events.
- Creation of notifications upon detecting symptoms of information security threats.
- Creation of alerts and incidents for processing information security threats.
- Displaying information about the status of the customer's infrastructure on the dashboard and in reports.
- Monitoring event sources.
- Device (asset) management — viewing information about assets, searching, adding, editing, and deleting assets, exporting asset information to a CSV file.

To perform its primary functions, KUMA may receive, store and process the following information:

- Information about devices on the corporate network.

  The KUMA Core server receives data if the corresponding integration is configured. You can add assets to KUMA in the following ways:

  - Import assets:
    - On demand from MaxPatrol.
    - On a schedule from Open Single Management Platform and KICS for Networks.
  - Create assets manually through the web interface or via the API.

KUMA stores the following device information:

- Technical characteristics of the device.
- Vulnerabilities, detected on an asset.
- Information specific to the source of the asset.

- Additional technical attributes of devices on the corporate network that the user specifies to send an incident to NCIRCC: IP addresses, domain names, URIs, email address of the attacked object, attacked network service, and port/protocol.
- Information about the organization: name, tax ID, address, email address for sending notifications.
- Active Directory information about organizational units, domains, users, and groups obtained as a result of querying the Active Directory network.

  The KUMA Core server receives this information if the corresponding integration is configured. To ensure the security of the connection to the LDAP server, the user must enter the server URL, the Base DN, connection credentials, and certificate in the KUMA Console.
- Information for domain authentication of users in KUMA: root DN for searching access groups in the Active Directory directory service, URL of the domain controller, certificate (the root public key that the AD certificate is signed with), full path to the access group of users in AD (distinguished name).
- Information contained in events from configured sources

  In the collector, the event source is configured, KUMA events are generated and sent to other KUMA services. Sometimes events can arrive first at the agent service, which relays events from the source to the collector. In addition, you can configure saving the address or a hostname of the server that aggregates the events.

  Also, Endpoint Agents send to OSMP information about events in accordance with the event data model.
- Information required for the integration of KUMA with other applications (Kaspersky Threat Lookup, Kaspersky CyberTrace, Open Single Management Platform, Kaspersky Industrial CyberSecurity for Networks, Kaspersky Automated Security Awareness Platform, Kaspersky Endpoint Detection and Response, Security Orchestration, Automation and Response, AI services: AI asset score and status, Kaspersky Investigation and Response Assistant).

  It can include certificates, tokens, URLs or credentials for establishing a connection with the other application, or other data necessary for the basic functionality of KUMA, for example, email. The user enters this data in the KUMA Console
- Information about sources from which event receipt is configured.

  It can include the source name, host name, IP address, the monitoring policy assigned to the source. The monitoring policy specifies the email address of the person responsible, to whom a notification will be sent if the policy is violated.
- User accounts: name, username, email address. The user can view their profile data in the KUMA Console.
- User profile settings:
  - User role in KUMA. A user can see their assigned roles.
  - Localization language, notification settings, display of non-printable characters.

    The user enters this data in the KUMA interface.

- List of asset categories in the **Assets** section, default dashboard, TV mode flag for the dashboard, SQL query for default events, default preset.

  The user specifies these settings in the corresponding sections of the KUMA Console.

- Data for domain authentication of users in KUMA:

  - Active Directory: root DN for searching access groups in the Active Directory directory service, URL of the domain controller, certificate (the root public key that the AD certificate is signed with), full path to the access group of users in AD (distinguished name).

  - Active Directory Federation Services: trusted party ID (KUMA ID in AD FS), URI for getting Connect metadata, URL for redirection from AD FS, and the AD FS server certificate.

  - FreeIPA: Base DN, URL, certificate (the public root key that was used to signed the FreeIPA certificate), custom integration credentials, connection credentials.

- Audit events

  KUMA automatically records audit events.

- KUMA log

  The user can enable extended logging in the KUMA Console. Log entries are stored on the user's device, no data is transmitted automatically.

- Information about the user accepting the terms and conditions of legal agreements with Kaspersky.

- Any information that the user enters in the KUMA interface.

The information listed above can find its way into KUMA in the following ways:

- The user enters information in the KUMA Console.

- KUMA services (agent or collector) receive data if the user has configured a connection to event sources.

- Through the KUMA REST API.

- Device information can be obtained using the utility from MaxPatrol.

The listed information is stored in the KUMA database (MongoDB, ClickHouse, SQLite). Passwords are stored in an encrypted form (the hash of the password is stored).

All of the information listed above can be transmitted to Kaspersky only in dump files, trace files, or log files of KUMA components, including log files created by the installer and utilities.

Dump files, trace files, and log files of KUMA components may contain personal and confidential information. Dump files, trace files, and log files are stored on the device in unencrypted form. Dump files, trace files, and log files are not automatically submitted to Kaspersky, but the administrator can manually submit this information to Kaspersky at the request of Technical Support to help troubleshoot KUMA problems.

Kaspersky uses the received data in anonymized form and only for general statistical purposes. Summary statistics are generated from the received raw data automatically and does not contain any personal or other confidential information. When new data accumulates, older data is erased (once a year). Summary statistics are stored indefinitely.

Kaspersky protects all received data in accordance with applicable law and Kaspersky policies. Data is transmitted over secure communication channels.

Quick start guide

The following scenarios are step-by-step walkthroughs from the purchase of Kaspersky Next XDR Expert to incident investigation and threat hunting.

Start with installation and initial setup of Kaspersky Next XDR Expert, then explore Kaspersky Next XDR Expert threat detection and hunting features, and then check out an example of an incident investigation workflow.

Deployment and initial setup of Kaspersky Next XDR Expert

Following this scenario, you can deploy Open Single Management Platform with all the components necessary for operation of the Kaspersky Next XDR Expert solution, and then perform the required preliminary configurations and integrations.

Prerequisites

Before you start, make sure that:

- You have a license key for Kaspersky Next XDR Expert and the compatible EPP applications.
- Your infrastructure meets the hardware and software requirements.

Stages

The main installation and initial setup scenario proceeds in stages:

1. **Deployment**

   Prepare your infrastructure for the deployment of Open Single Management Platform and all the required components for Kaspersky Next XDR Expert, and then deploy the solution by using the Kaspersky Deployment Toolkit utility.

2. **Activation**

   Activate the Kaspersky Next XDR Expert solution under your license.

3. **Configuring multitenancy**

   If necessary, you can use the multitenancy features:

   1. Plan and create the required hierarchy of tenants.
   2. Create the matching hierarchy of Administration Servers in Open Single Management Platform.
   3. Bind tenants to the corresponding Administration Servers.
   4. Create user accounts for all Kaspersky Next XDR Expert users, and then assign roles.

4. **Adding assets**

   The devices in your infrastructure that must be protected are represented as assets in Kaspersky Next XDR Expert. Open Single Management Platform allows you to discover the devices in your network and manage their protection. You will also be able to add assets manually or import them from other sources during stage 8.

   User accounts are also represented as assets in Kaspersky Next XDR Expert. Make sure to configure the integration with Active Directory during stage 9, to enable the display of affected user accounts in the related events, alerts, and incidents.

5. **Adding users and assigning roles**

   Assign roles to the user accounts, to define their access rights to various Kaspersky Next XDR Expert features depending on their tasks.

6. **Connecting to an SMTP server**

   Configure the connection to an SMTP server for email notifications about events occurring in Kaspersky Next XDR Expert.

7. **Installing endpoint protection applications and solutions**

   Kaspersky Next XDR Expert works with events received from security applications installed on your assets. Check the list of compatible Kaspersky applications and solutions. You can use Open Single Management Platform to deploy Kaspersky applications on the devices in your infrastructure.

   Ensure that endpoint protection applications are integrated with Kaspersky Anti Targeted Attack Platform. For example, if you use Kaspersky Endpoint Security on your assets, refer to one of the following Help documentations to learn how to configure integration with KATA:

   - Kaspersky Endpoint Security for Windows
   - Kaspersky Endpoint Security for Linux
   - Kaspersky Endpoint Security for Mac

8. **Configuring event sources, storage, and correlation**

   Specify where the events must be received from, and how they must be stored and processed:

   1. Log in to the KUMA Console.
   2. Set up integration of Kaspersky Unified Monitoring and Analysis Platform and Kaspersky Security Center.
   3. **Import assets from Open Single Management Platform**.

      All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

      KUMA imports the following device information received from Kaspersky Security Center Network Agents:

      - Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
      - Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Click **Import**, and then select one of the following actions:
   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.
   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.
3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

   If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.
4. Click the **Import** button.

The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

4. Add assets manually or import them from other sources (optional action).
5. Configure the event sources to specify where you want to receive the events from.
6. Create a storage for events.
7. Create collectors for receiving, processing (normalizing), and transmitting the events.
8. Create correlators for initial analysis of normalized events and their further processing.

   During the correlator creation, you can create correlation rules to define the rules of processing and responding to the events.You can also import the previously saved correlation rules or use the ready-made set of correlation rules provided with the Kaspersky Next XDR Expert solution. After the correlator is created, you can link correlation rules to the correlator, if needed.

   We strongly recommended configuring the exclusions on this stage, to avoid false positives and irrelevant data.

9. **Configuring the integrations**

   Configure the integration of Kaspersky Next XDR Expert with Active Directory and with other Kaspersky solutions, to extend its possibilities and to enrich data available for incident investigation.

   1. Integration with Active Directory (strongly recommended).
   2. Integration with KATA/EDR (license is required).
   3. Integration with Kaspersky CyberTrace (optional integration; license is required).

4. Integration with Kaspersky TIP (optional integration; license is required) or Kaspersky Open TIP.

5. Integration with Kaspersky Automated Security Awareness Platform (optional integration; license is required).

10. **Configuring updates**

    Create the Download updates to the Administration Server repository task.

11. **Verify correctness of configuration**

    Use the EICAR test file on one of the assets. If the initial setup was performed correctly and the necessary correlation rules were configured, this event will trigger creation of an alert in the alerts list.

After the initial setup is complete, events from the protected assets will be received and processed by Kaspersky Next XDR Expert, and an alert will be created when a correlation rule is triggered.

Verifying correctness of the Kaspersky Next XDR Expert configuration

You can use the EICAR test virus on one of the assets, to ensure that Kaspersky Next XDR Expert is deployed and configured correctly. If the initial setup was performed correctly and the necessary correlation rules were configured, the correlation event will trigger the creation of an alert in the alerts list.

To verify correctness of the Kaspersky Next XDR Expert configuration:

1. Create a new correlator in KUMA section of OSMP Console.

   When creating the correlator, do not specify parameters in the **Correlation** section.

2. Import correlation rules from the SOC Content package to obtain the predefined correlation rules used to detect the EICAR test virus.

3. Specify the correlation rule for the created correlator.

   You can use one of the following methods to specify the correlation rule:

   - Link the predefined correlation rule to the created correlator:

     a. Go to **Resources**, click **Correlation rules**, and then select the tenant to which the correlation rule will be applied.

     b. In the list of the predefined correlation rules, select the **R077_02_KSC.Malware detected** rule to detect events from Kaspersky Security Center.

     c. Click **Link to correlator**, and then select the created correlator to link the selected correlation rule to the correlator.

- Create the correlation rule with the predefined filters manually:

  a. Open the created correlator settings, go to the **Correlation section**, and then click **Add**.

  b. In the **Create correlation rule** window, on the **General** tab, set the following parameters, as well as other rule parameters:

     - **Kind**: simple.

     - **Propagated fields**: DestinationAddress, DestinationHostName, DestinationAccountID, DestinationAssetID, DestinationNtDomain, DestinationProcessName, DestinationUserName, DestinationUserID, SourceAccountID, SourceUserName.

  c. Go to **Selectors** → **Settings**, and then specify the expression to filter the required events:

     - In builder mode, add the **f: KSC events**, **f: KSC virus found**, and **f: Base events** filters with the AND operator.

     - Alternatively, you can specify this expression in the source code mode as follows:

       ```
       filter='b308fc22-fa79-4324-8fc6-291d94ef2999'
       AND filter='a1bf2e45-75f4-45c1-920d-55f5e1b8843f'
       AND filter='1ffa756c-e8d9-466a-a44b-ed8007ca80ca'
       ```

  d. In the **Actions** section of the correlation rule settings, select only the **Output** check box (the **Loop to correlator** and **No alert** check boxes must be cleared). In this case, when the EICAR test virus is detected, a correlation event will be created and an alert will be created in the alert list of Kaspersky Next XDR Expert.

  e. Click **Create new** to save the correlation rule settings linked to the correlator.

4. Create, and then configure, a collector in KUMA section of OSMP Console for receiving information about Administration Server events from an MS SQL database.

   Alternatively, you can use the predefined [OOTB] KSC SQL collector.

5. In the **Routing** section of the collector settings, set **Type** to **correlator**, and then specify the created correlator in the **URL** field, to forward the processed events to it.

6. Install Network Agent and the endpoint protection application (for example, Kaspersky Endpoint Security) on an asset of your organization network. Ensure that the asset is connected to Administration Server.

7. Place the EICAR test file on the asset, and then detect the test virus by using the endpoint protection application.

After that, Administration Server will be notified about the event on the asset. This event will be forwarded to the KUMA component, transformed to the correlation event, and then this event will trigger creation of an alert in the alerts list in Kaspersky Next XDR Expert. If the alert has been created, it means that Kaspersky Next XDR Expert is working correctly.

Using the threat monitoring, detection and hunting features

After you have installed and configured Kaspersky Next XDR Expert, you can use Kaspersky Next XDR Expert features for monitoring the security of your infrastructure, investigating security incidents, automating workflows and proactive searching for threats:

- **Using dashboard and customizing widgets**

  The **Detection and response** tab of the dashboard can contain widgets that display information about detected and registered alerts and incidents, and response actions to them. You can use and customize the preconfigured layouts of widgets for your dashboard or create new layouts and widgets.

  Open Single Management Platform also provides various security monitoring and reporting tools.

- **Using reports**

  You can configure the generation of reports in Kaspersky Unified Monitoring and Analysis Platform to receive the required summary data according to the specified schedule.

- **Using threat hunting**

  You can use threat hunting tools to analyze events to search for threats and vulnerabilities that have not been detected automatically. Threat hunting can be used both for alert and incident investigation and for proactive search for threats.

- **Using playbooks**

  You can use playbooks to automate response to alerts and incidents according to the specified algorithm. There are a number of predefined playbooks that you can launch in various operation modes. You can create custom playbooks.

Example of incident investigation with Kaspersky Next XDR Expert

This scenario represents a sample workflow of an incident investigation.

Incident investigation proceeds in stages:

1. **Assigning an alert to a user**

   You can assign an alert to yourself or to another user.

2. **Checking if the triggered correlation rule matches the data of the alert events**

   View the information about the alert and make sure that the alert event data matches the triggered correlation rule. If the alert is a false positive, edit the correlation rule.

3. **Analyzing alert information**

   Analyze the information about the alert to determine what data is required for further analysis of the alert.

4. **Manual enrichment**

   Launch the available solutions for additional enrichment of an event (for example, Kaspersky TIP).

5. **False positive check**

   Make sure that the activity that triggered the correlation rule is abnormal for the organization IT infrastructure.

6. **Incident creation**

   If steps from 3 to 5 reveal that the alert requires investigation, you can create an incident or link the alert to an existing incident.

   You can also merge incidents.

7. **Incident management**

   You can manage incidents by changing incident statuses and priorities, and by assigning them to other analyst.

   You can also manage incident SLAs, types, and workflows. In addition, you can track the effort spent on incident-related work.

8. **Investigation**

   This step includes viewing information about the assets, user accounts, and alerts related to the incident. You can use the investigation graph and threat hunting tools to get additional information.

9. **Searching for related assets**

   You can view the alerts that occurred on the assets related to the incident.

10. **Searching for related events**

    You can expand your investigation scope by searching for events of related alerts.

11. **Viewing the history of the incident**

    You can view the incident's change log and response history.

12. **Commenting**

    You can leave comments with the information necessary for the investigation of the incident.

13. **Response**

    You can perform response actions manually.

14. **Responding with playbooks**

    You can respond to an incident by launching playbooks manually.

15. **Closing the incident**

    After taking measures to clean up the traces of the attacker's presence from the organization's IT infrastructure, you can close the incident.

Deployment of Kaspersky Next XDR Expert

Following this scenario, you can prepare your infrastructure for the deployment of Open Single Management Platform and all the required components for Kaspersky Next XDR Expert (including Kaspersky Endpoint Detection and Response functionality), prepare the configuration file containing the installation parameters, and deploy the solution by using the Kaspersky Deployment Toolkit utility (hereinafter referred to as KDT).

Before you deploy Open Single Management Platform and Kaspersky Next XDR Expert components, we recommend reading the Hardening Guide.

The deployment scenario proceeds in stages:

1. **Selecting the option for deploying Kaspersky Next XDR Expert**

   Select the configuration of Kaspersky Next XDR Expert that best suits your organization. Multi-node and single-node deployment are available.

2. **Downloading the distribution package with the Kaspersky Next XDR Expert components**

   The distribution package contains the following components:

   - Transport archive with the Kaspersky Next XDR Expert components and End User License Agreements for Kaspersky Next XDR Expert and KDT.
   - Archive with the KDT utility, and templates of the configuration file and KUMA inventory file.

3. **Installing a database management system (DBMS)**

   For multi-node deployment, manually install the DBMS on the separated server that will not be included the Kubernetes cluster.

   For single-node deployment, manually install the DBMS on the target host before the Kaspersky Next XDR Expert deployment. In this case, the DBMS and Kaspersky Next XDR Expert components are installed on the same target host, but the DBMS is not included in the Kubernetes cluster.

4. **Preparing the administrator and target hosts**

   Based on the selected deployment scheme, define the number of target hosts on which you will deploy the Kubernetes cluster and the Kaspersky Next XDR Expert components included in this cluster. Prepare the selected administrator host and target hosts for deployment of Kaspersky Next XDR Expert.

   How-to instructions:

   - Multi-node deployment: Preparing the administrator and target hosts
   - Single-node deployment: Preparing the administrator and target hosts

5. **Preparing the KUMA hosts for installing KUMA services outside the Kubernetes cluster (optional)**

   KUMA services (collectors, correlators, and storages) can be installed entirely inside the Kubernetes cluster, entirely on separate hosts outside the cluster, or partially inside and partially outside the cluster.

   If you want to install KUMA services entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the cluster, prepare the KUMA target hosts for installing KUMA services.

   How-to instruction: Preparing KUMA hosts for installing KUMA service outside the Kubernetes cluster

6. **Preparing the KUMA inventory file for installing KUMA services outside the Kubernetes cluster (optional)**

   If you want to install KUMA services entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the cluster, prepare the KUMA inventory file in the YAML format. The KUMA inventory file contains parameters for installation of the KUMA services outside the Kubernetes cluster.

   How-to instruction: Preparing the KUMA inventory file for installing KUMA services outside the Kubernetes cluster

7. **Preparing the configuration file**

   Prepare the configuration file in the YAML format. The configuration file contains the list of target hosts for deployment and a set of installation parameters of the Kaspersky Next XDR Expert components.

   If you deploy Kaspersky Next XDR Expert on a single-node, use the configuration file that contains the installation parameters specific for single-node deployment.

   How-to instructions:

   - Multi-node deployment: Specifying the installation parameters
   - Single-node deployment: Specifying the installation parameters

   You can fill out the configuration file template manually; or use the Configuration wizard to specify the installation parameters that are required for the Kaspersky Next XDR Expert deployment, and then generate the configuration file.

   How-to instruction: Specifying the installation parameters by using the Configuration wizard

8. **Deployment of Kaspersky Next XDR Expert**

   Deploy Kaspersky Next XDR Expert by using KDT. KDT automatically deploys the Kubernetes cluster within which the Kaspersky Next XDR Expert components and other infrastructure components are installed.

   How-to instruction: Installing Kaspersky Next XDR Expert

9. **Installing the KUMA services**

   Install the KUMA services (collectors, correlators, and storages).

   KUMA services can be installed entirely inside the Kubernetes cluster, entirely on separate hosts outside the cluster, or partially inside and partially outside the Kubernetes cluster.

   How-to instruction: Installing KUMA services outside the Kubernetes cluster

   How-to-instruction: Installing KUMA services inside the Kubernetes cluster

10. **Adding a secondary Administration Server**

    A secondary Administration Server is required for further integration of Endpoint Agents.

    How-to instructions: Adding a secondary Administration Server

11. **Installing and integrating Endpoint Agents**

    Install Endpoint Agents according to Endpoint Agent documentation:

    - Kaspersky Endpoint Security for Windows
    - Kaspersky Endpoint Security for Linux
    - Kaspersky Endpoint Security for Mac

    Configure integration with Endpoint Agents.

12. **Installing and integrating Sandbox**

    Install Sandbox, configure it, and then install virtual machines in it.

    Configure integration with Sandbox.

Hardening Guide

The Hardening Guide is intended for professionals who deploy and administer Kaspersky Next XDR Expert, as well as for those who provide technical support to organizations that use Kaspersky Next XDR Expert.

The Hardening Guide describes recommendations and features of configuring Kaspersky Next XDR Expert and its components, aimed to reduce the risks of its compromise.

The Hardening Guide contains the following information:

- Preparing the infrastructure for the Kaspersky Next XDR Expert deployment
- Configuring a secure connection to Kaspersky Next XDR Expert
- Configuring accounts to access Kaspersky Next XDR Expert
- Managing protection of Kaspersky Next XDR Expert
- Managing protection of client devices
- Configuring protection for managed applications
- Transferring information to third-party applications

Before you start to deploy Kaspersky Next XDR Expert, we recommend reading the Hardening Guide.

Managing infrastructure of Kaspersky Next XDR Expert

This section describes the general principle of using the minimum required number of applications for the function of the operating system and Kaspersky Next XDR Expert. This section also describes the principle of least privilege, which boils down to the concept of Zero Trust.

Managing operating system accounts

To work with a Kubernetes cluster by using KDT, we recommend creating a separate user with minimal privileges. The optimal way is to implement management of user accounts of the operating system by using LDAP, with the ability to revoke user rights through LDAP. For the specific implementation of user revocation and blocking, see the user/administrator guide in your LDAP solution. We recommend using a password of at least 18 characters or a passphrase with any delimiters of at least 4 words to authenticate the user. You can also use a physical means of authentication (for example, token).

For more information, refer to OpenLDAP Software documentation.

Example

Distributive	User creation command	Minimum set of groups/privileges
Ubuntu / Debian / Astra / Atlant	`useradd "$1" --shell /bin/bash --create-home --password "$(openssl passwd -1 "$2")"`	`usermod -aG sudo,root "$1"` Adding to the `sudo` group allows user to execute `sudo` without a password.

We also recommend protecting the user home directory and all nested directories in such a way that only the user has access to them. Other users and the user group must not have rights to the home directory.

Example

In this example, an `example_user` user has a document directory located at `~/Documents`. By default (when creating a user via `useradd/adduser` and `umask 022` commands), directories are created with permissions `755`. To ensure that only this user has access to the root folder and all subfolders, the `umask` is changed to `077` and the permissions are explicitly set to `700`. Below is an example of commands that can be used to verify that the permissions are already set correctly (only the owner has access).

To show permissions, owner, and group of a directory:
`stat -c "%A %U %G" /home/example_user/Documents`
Expected command output if permissions are granted correctly:
`drwx------ example_user example_user`
If the permissions do not meet the requirements, set the permissions to `700` for the root directory and all subdirectories:
`chmod -R 700 /home/example_user/Documents`
Ensure the owner is the correct user:
`chown -R example_user:example_user /home/example_user/Documents`

We recommend not granting the execute permission for the .ssh, .kube, .config, and .kdt directories, and all the contained files in these directories in the user's home directory.

Example

Display permissions for the most sensitive directories –show the mode, owner, and path for each of the specified directories:

```
find ~/Documents -type d \( -name .ssh -o -name .kube -o -name .config -o -name .kdt
\) \

-exec stat -c "%A %U %G %n" {} \;
```

Expected command output:

```
drwx------ example_user example_user /home/example_user/Documents/.ssh

drwx------ example_user example_user /home/example_user/Documents/.kube

drwx------ example_user example_user /home/example_user/Documents/.config

drwx------ example_user example_user /home/example_user/Documents/.kdt
```

`drwx------` represents permissions `700`: only the owner can read, write, and execute (in the case of directories, "execute" is the permission to enter a directory, but not to run software).

If extra execute bits are detected, remove them from files:

```
find ~/Documents -type f \( -path "*/.ssh/*" -o -path "*/.kube/*" -o -path
"*/.config/*" -o -path "*/.kdt/*" \) -exec chmod a-x {} \;
```

If necessary, set stricter permissions (for example, `600`):

```
find ~/Documents -type f \( -path "*/.ssh/*" -o -path "*/.kube/*" -o -path
"*/.config/*" -o -path "*/.kdt/*" \) -exec chmod 600 {} \;
```

Package management of the operating system

We recommend using the minimum set of applications required for the function of KDT and Kaspersky Next XDR Expert. For example, you do not need to use a graphical user interface for working in the Kubernetes cluster, so we recommend not installing graphical packages. If packages are installed, we recommend removing these packages, including graphical servers such as Xorg or Wayland.

We recommend regularly installing security updates for the system software and the Linux kernel. We also recommend enabling automatic updates as follows:

- For operating systems with the atp package manager:

  /etc/apt/apt.conf.d/50unattended-upgrades

  ```
  Unattended-Upgrade::Allowed-Origins {
      "${distro_id}:${distro_codename}-security";

      "${distro_id}ESMApps:${distro_codename}-apps-security";
      "${distro_id}ESM:${distro_codename}-infra-security";
  };
  ```

- For operating systems with the rp, dnf, and yum package managers:

  /etc/dnf/automatic.conf

  ```
  [commands]
  #  What kind of upgrade to perform:
  # default                            = all available upgrades
  # security                           = only the security upgrades
  upgrade_type = default

  # Whether updates should be downloaded when they are available, by
  # dnf-automatic.timer. notifyonly.timer, download.timer and
  # install.timer override this setting.
  download_updates = yes

  # Whether updates should be applied when they are available, by
  # dnf-automatic.timer. notifyonly.timer, download.timer and
  # install.timer override this setting.
  apply_updates = no
  ```

Operating system security settings

The Linux kernel security settings can be enabled in the `/etc/sysctl.conf` file or by using the `sysctl` command. The recommended Linux kernel security settings are listed in the `/etc/sysctl.conf` file snippet:

```
/etc/sysctl.conf

# Disable execshield
kernel.randomize_va_space=2
# Enable IP spoofing protection
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.default.rp_filter=1
# Ignore broadcast network requests
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.icmp_ignore_bogus_error_responses=1
# Enable logging of network spoofing packages
net.ipv4.conf.all.log_martians=1
# Hide the kernel pointers
kernel.kptr_restrict=1
# Restrict access to the kernel logs
kernel.dmesg_restrict = 1
# Prohibit the kernel profiling for unprivileged users
kernel.perf_event_paranoid=3
# Increasing the ASLR entropy bits
vm.mmap_rnd_bits=32
vm.mmap_rnd_compat_bits=16
```

We recommend restricting access to the PID. This will reduce the possibility of one user tracking the processes of another user. You can restrict access to the PID while mounting the `/proc` file system, for example, by adding the following line to the `/etc/fstab` file:

```
proc /proc proc nosuid,nodev,noexec,hidepid=2,gid=proc 0 0
```

If the operating system processes are managed by using the `systemd` system, the `systemd-logind` service can still monitor the processes of other users. In order for user sessions to work correctly in the `systemd` system, you need to create the `/etc/systemd/system/systemd-logind.service.d/hidepid.conf` file, and then add the following lines to it:

```
[Service]
SupplementaryGroups=proc
```

Since some systems may not have the `proc` group, we recommend adding the `proc` group in advance.

We recommend turning off the **`ctrl`**+**`alt`**+**`del`** key combination, to prevent an unexpected reboot of the operating system by using the `systemctl mask ctrl-alt-del.target` command.

We recommend prohibiting authentication of privileged users (root users) to establish a remote user connection.

Example

Below is a set of commands that simultaneously prevent root from logging in over the network and exclude it from LDAP authentication.

```
# 1. SSH

sudo sed -i 's/^#\?PermitRootLogin .*/PermitRootLogin no/' /etc/ssh/sshd_config
```

```
sudo systemctl restart sshd

# 2. Local account

sudo passwd -l root

sudo usermod -s /usr/sbin/nologin root

# 3. Example of LDAP bind, the file path may differ

LDAP_CONF="/etc/ksc/ldap_connection.json"

if [ -f "$LDAP_CONF" ]; then

sudo jq '.ldap.bindDn = "cn=admin,dc=example,dc=com"' "$LDAP_CONF" | sudo tee
"$LDAP_CONF" > /dev/null

sudo systemctl restart ksc-orchestrator

fi

# 4. Sudo without NOPASSWD

sudo sed -i '/^root.*NOPASSWD/d' /etc/sudoers
```

We recommend using a firewall to limit network activity. For more information about the ports and protocols used, refer to Ports used by Kaspersky Next XDR Expert.

Example

Action	ufw (Ubuntu/Debian)	firewalld (RHEL/CentOS)	iptables
Check status	ufw status	firewall-cmd --list-all	iptables -L -n
Check port	ss -tuln | grep 443	ss -tuln | grep 443	ss -tuln | grep 443
External audit	nc -zv IP 443	nc -zv IP 443	nc -zv IP 443
Open port	ufw allow 443/tcp	firewall-cmd --add-port=443/tcp --permanent + --reload	iptables -A INPUT -p tcp --dport 443 -j ACCEPT
Close port	ufw deny 443/tcp	firewall-cmd --remove-port=443/tcp --permanent + --reload	iptables -A INPUT -p tcp --dport 443 -j DROP
Save rules	Auto	Auto	iptables-save > /etc/iptables/rules.v4

We recommend enabling `auditd`, to simplify the investigation of security incidents. For more information about enabling telemetry redirection, refer to Setting up receiving Auditd events.

We recommend regularly backing up the following configurations and data directories:

- Administration host: `~/kdt`
- Target hosts: `/etc/k0s/`, `/var/lib/k0s`

Also we recommend encrypting these backups.

Hardening guides for various operating systems and for DBMS

If you need to configure the security settings of your operating system and software, you can use the recommendations provided by Center for Internet Security (CIS).

If you use the Astra Linux operating system, refer to the security recommendations that can be applied to your Astra Linux version.

If you need to configure security settings of PostgreSQL, use the server administration recommendations from the official PostgreSQL documentation.

Connection safety

Strict TLS settings

We recommend using TLS protocol version 1.2 and later, and restricting or prohibiting insecure encryption algorithms.

You can configure encryption protocols (TLS) used by Administration Server. Please note that at the time of the release of a version of Kaspersky Next XDR Expert, the encryption protocol settings are configured by default to ensure secure data transfer.

Restricting access to the Kaspersky Next XDR Expert database

We recommend restricting access to the Kaspersky Next XDR Expert database. For example, grant access only from devices with Kaspersky Next XDR Expert deployed. This reduces the likelihood of the Kaspersky Next XDR Expert database being compromised due to known vulnerabilities.

You can configure the parameters according to the operating instructions of the used database, as well as provide closed ports on firewalls.

Accounts and authentication

Using two-step verification with Kaspersky Next XDR Expert

Kaspersky Next XDR Expert provides two-step verification for users, based on the RFC 6238 standard (TOTP: Time-Based One-Time Password algorithm).

When two-step verification is enabled for your own account, every time you log in to Kaspersky Next XDR Expert through a browser, you enter your user name, password, and an additional single-use security code. To receive a single-use security code, you must install an authenticator app on your computer or your mobile device.

There are both software and hardware authenticators (tokens) that support the RFC 6238 standard. For example, software authenticators include Google Authenticator, Microsoft Authenticator, FreeOTP.

We strongly do not recommend installing the authenticator app on the same device from which the connection to Kaspersky Next XDR Expert is established. You can install an authenticator app on your mobile device.

Using two-factor authentication for an operating system

We recommend using multi-factor authentication (MFA) on devices with Kaspersky Next XDR Expert deployed, by using a token, a smart card, or other method (if possible).

Prohibition on saving the administrator password

If you use Kaspersky Next XDR Expert through a browser, we do not recommend saving the administrator password in the browser installed on the user device.

Authentication of an internal user account

By default, the password of an internal user account of Kaspersky Next XDR Expert must comply with the following rules:

- The password must be 8 to 256 characters long.
- The password must contain characters from at least three of the groups listed below:
  - Uppercase letters (A-Z)
  - Lowercase letters (a-z)
  - Numbers (0-9)
  - Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)
- The password must not contain any whitespaces, Unicode characters, or the combination of "." and "@", when "." is placed before "@".

The maximum number of allowed attempts to enter a password is 5. This number is fixed and cannot be changed.

The user can enter an invalid password a limited number of times. After the limit is reached, the user account is blocked for one hour.

Restricting the assignment of the Main Administrator role

The user is assigned the Main Administrator role in the access control list (ACL) of Kaspersky Next XDR Expert. We do not recommend assigning the Main Administrator role to a large number of users.

Configuring access rights to application features

We recommend using flexible configuration of access rights to the features of Kaspersky Next XDR Expert for each user or group of users.

Role-based access control allows the creation of standard user roles with a predefined set of rights and the assignment of those roles to users depending on their scope of duties.

The main advantages of the role-based access control model:

- Ease of administration
- Role hierarchy
- Least privilege approach
- Segregation of duties

You can assign built-in roles to certain employees based on their positions, or create completely new roles.

While configuring roles, pay attention to the privileges associated with changing the protection state of the device with Kaspersky Next XDR Expert and remote installation of third-party software:

- Managing administration groups.
- Operations with Administration Server.
- Remote installation.
- Changing the parameters for storing events and sending notifications.

  This privilege allows you to set notifications that run a script or an executable module on the device with OSMP when an event occurs.

Separate account for remote installation of applications

In addition to the basic differentiation of access rights, we recommend restricting the remote installation of applications for all accounts (except for the Main Administrator or another specialized account).

We recommend using a separate account for remote installation of applications. You can assign a role or permissions to the separate account.

Regular audit of all users and users' actions

We recommend conducting a regular audit of all users on devices with Kaspersky Next XDR Expert deployed. This allows you to respond to certain types of security threats associated with the possible compromise of a device.

Also, you can track the users' actions, such as connecting to and disconnecting from Administration Server, connecting to Administration Server with an error, and objects modification (for objects that support revision management).

Managing protection of Kaspersky Next XDR Expert

Selecting protection software of Kaspersky Next XDR Expert

Depending on the type of the Kaspersky Next XDR Expert deployment and the general protection strategy, select the application to protect devices with Kaspersky Next XDR Expert deployed and the administrator host.

If you deploy Kaspersky Next XDR Expert on dedicated devices, we recommend selecting the Kaspersky Endpoint Security application to protect devices with Kaspersky Next XDR Expert deployed and the administrator host. This allows applying all available technologies to protect these devices, including behavioral analysis modules.

If Kaspersky Next XDR Expert is deployed on devices that exists in the infrastructure and has previously been used for other tasks, we recommend considering the following protection software:

- Kaspersky Industrial CyberSecurity for Nodes. We recommend installing this application on devices that are included in an industrial network. Kaspersky Industrial CyberSecurity for Nodes is an application that has certificates of compatibility with various manufacturers of industrial software.

- Recommended security applications. If Kaspersky Next XDR Expert is deployed on devices with other software, we recommend taking into account the recommendations from that software vendor on the compatibility of security applications (there may already be recommendations for selecting a security solution, and you may need to configure the trusted zone).

Protection modules

If there are no special recommendations from the vendor of the third-party software installed on the same devices as Kaspersky Next XDR Expert, we recommend activating and configuring all available protection modules (after checking the operation of these protection modules for a certain time).

Configuring the firewall of devices with Kaspersky Next XDR Expert

On devices with Kaspersky Next XDR Expert deployed, we recommend configuring the firewall to restrict the number of devices from which administrators can connect to Kaspersky Next XDR Expert through a browser.

By default,

Kaspersky Next XDR Expert uses port 443 to log in through a browser. We recommend restricting the number of devices from which Kaspersky Next XDR Expert can be managed by using this port.

Managing protection of client devices

Restricting of adding license keys to installation packages

Installation packages can be published through Web Server, which is included in Kaspersky Next XDR Expert. If you add a license key to the installation package that is published on Web Server, the license key will be available for all users to read.

To avoid compromising the license key, we do not recommend adding license keys to installation packages.

We recommend using automatic distribution of license keys to managed devices, deployment through the *Add license key* task for a managed application, and adding an activation code or a key file manually to the devices.

Automatic rules for moving devices between administration groups

We recommend restricting the use of automatic rules for moving devices between administration groups.

If you use automatic rules for moving devices, this may lead to propagation of policies that provide more privileges to the moved device than the device has before relocation.

Also, moving a client device to another administration group may lead to propagation of policy settings. These policy settings may be undesirable for distribution to guest and untrusted devices.

This recommendation does not apply for one-time initial allocation of devices to administration groups.

Security requirements for distribution points and connection gateways

Devices with Network Agent installed can act as a distribution point and perform the following functions:

- Distribute updates and installation packages received from Kaspersky Next XDR Expert to client devices within the group.
- Perform remote installation of third-party software and Kaspersky applications on client devices.
- Poll the network to detect new devices and update information about existing ones. The distribution point can use the same methods of device detection as Kaspersky Next XDR Expert.

Placing distribution points on the organization's network used for:

- Reducing the load on Kaspersky Next XDR Expert
- Traffic optimization
- Providing Kaspersky Next XDR Expert with access to devices in hard-to-reach parts of the network

Taking into account the available capabilities, we recommend protecting devices that act as distribution points from any type of unauthorized access (including physically).

Restricting automatic assignment of distribution points

To simplify administration and keep the network operability, we recommend using automatic assignment of distribution points. However, for industrial networks and small networks, we recommend that you avoid assigning distribution points automatically, since, for example, the private information of the accounts used for pushing remote installation tasks, can be transferred to distribution points by means of the operating system.

For industrial networks and small networks, you can manually assign devices to act as distribution points.

You can also view the Report on activity of distribution points.

Event transfer to third-party systems

This section describes the specifics of transferring security issues found on client devices to third-party systems.

Monitoring and reporting

For timely response to security issues, we recommend configuring the monitoring and reporting features.

Export of events to SIEM systems

For fast detection of security issues before significant damage occurs, we recommend using event export in a SIEM system.

Email notifications of audit events

For timely response to emergencies, we recommend configuring Administration Server to send notifications about the audit events, critical events, failure events, and warnings that it publishes.

Since these events are intra-system events, a small number of them can be expected, which is quite applicable for mailing.

Configuring protection for managed applications

Managed application policies

We recommend creating a policy for each type of the used applications and for all components of Kaspersky Next XDR Expert (Network Agent, Kaspersky Endpoint Security for Windows, Kaspersky Endpoint Agent, and others). This policy must be applied to all managed devices (the root administration group) or to a separate group to which new managed devices are automatically moved according to the configured movement rules.

Specifying the password for disabling protection and uninstalling the application

We strongly recommend enabling password protection to prevent intruders from disabling or uninstalling Kaspersky security applications. On platforms where password protection is supported, you can set the password, for example, for Kaspersky Endpoint Security, Network Agent, and other Kaspersky applications. After you enable password protection, we recommend locking the corresponding settings by closing the "lock."

Using Kaspersky Security Network

In all policies of managed applications and in the Kaspersky Next XDR Expert properties, we recommend enabling the use of Kaspersky Security Network (KSN) and accepting the KSN Statement. When you update Kaspersky Next XDR Expert, you can accept the updated KSN Statement. In some cases, when the use of cloud services is prohibited by law or other regulations, you can disable KSN.

Regular scan of managed devices

For all device groups, we recommend creating a task that periodically runs a full scan of devices.

Discovering new devices

We recommend properly configuring device discovery settings: set up integration with domain controllers and specify IP address ranges for discovering new devices.

For security purposes, you can use the default administration group that includes all new devices and the default policies affecting this group.

Deployment schemes

There are two options for deploying Kaspersky Next XDR Expert (including Kaspersky Endpoint Detection and Response functionality): on multiple nodes or on a single node within a Kubernetes cluster. Before you start, we recommend that you familiarize yourself with the available deployment schemes, and then choose the one that best meets your organization's requirements. You can also refer to the sizing guide, which outlines hardware requirements and recommends deployment options based on the number of devices in your organization.

Depending on the deployment option you choose, the following hosts may be required for Kaspersky Next XDR Expert to function properly:

- **Administrator host**

  The administrator host is a physical or virtual machine that is used to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. Since KDT runs on the administrator host, this host must meet the requirements for KDT.

  If the administrator host is not included in the cluster, it will be used only to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. If the administrator host is included in the cluster, it will also act as a target host that is used for operation of Kaspersky Next XDR Expert components. In this case, the host must meet the requirements for target hosts.

- **Target hosts**

  Target hosts are the physical or virtual machines that are used to deploy Kaspersky Next XDR Expert. The following target hosts are used:

  - Target hosts for installing the Kaspersky Next XDR Expert components inside the cluster

    The hosts that are included in the Kubernetes cluster and between which the workload is distributed. The target hosts must meet the requirements for the selected deployment option (multi-node or single node deployment).

    The KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster, entirely on separate hosts outside the cluster, or partially inside and partially outside the Kubernetes cluster. In this case, hosts must meet the hardware, software, and installation requirements that are necessary for installing the KUMA services.

    You can use one of the target hosts as the administrator host to run the Kaspersky Next XDR Expert deployment. In this case, this host must also meet the requirements for KDT.

  - KUMA target hosts for installing the KUMA services (optional)

    The KUMA services (collectors, correlators, and storages) can be installed on target hosts on separate hosts outside the cluster. The number of KUMA target hosts depends on the amount of events that Kaspersky Next XDR Expert has to process.

KUMA target hosts must meet the hardware, software, and installation requirements that are necessary for installing the KUMA services.

- **DBMS host (multi-node deployment only)**

  If you want to receive telemetry from Kaspersky Anti Targeted Attack Platform and manage threat response actions on assets connected to Kaspersky Endpoint Detection and Response servers, you can install and configure Kaspersky Anti Targeted Attack Platform with Kaspersky Endpoint Detection and Response. Kaspersky Anti Targeted Attack Platform is a standalone solution that must be installed on a separate server that is not included in the Kubernetes cluster. For details about KATA deployment scenarios, refer to the KATA documentation.

  The host for installing the DBMS is a separate server located outside the Kubernetes cluster. This host must meet the requirements for the database node.

Multi-node deployment

In the multi-node deployment, Kaspersky Next XDR Expert components are installed across several worker nodes in the Kubernetes cluster. If one node fails, the cluster can restore component operation on another node.

In this configuration, you need at least five hosts:

- One host for the DBMS
- Four target hosts for the Kubernetes cluster and Kaspersky Next XDR Expert components:
  - One host serving as both the administrator and a target host
  - Three target hosts

  If you want to use the administrator host located outside the cluster, you need to allocate one additional host for deployment.

  In order to install KUMA services outside the cluster, you must allocate at least one additional KUMA target host.

Single-node deployment

In the single-node deployment, all Kaspersky Next XDR Expert components are installed on a single node within the Kubernetes cluster. This option is suitable for solutions that require fewer computing resources.

In this configuration, at least one target host is required to install the Kubernetes cluster, Kaspersky Next XDR Expert components, and the DBMS. This host serves both as the administrator and target host. If you prefer to use an administrator host located outside the cluster, you must allocate an additional host for deployment.

If you want to install KUMA services outside the cluster, at least one additional KUMA target host must be allocated.

In this configuration, the DBMS does not require a separate node. However. it should be installed manually on the target host prior to deploying Kaspersky Next XDR Expert.

Deploying the hierarchy of single nodes

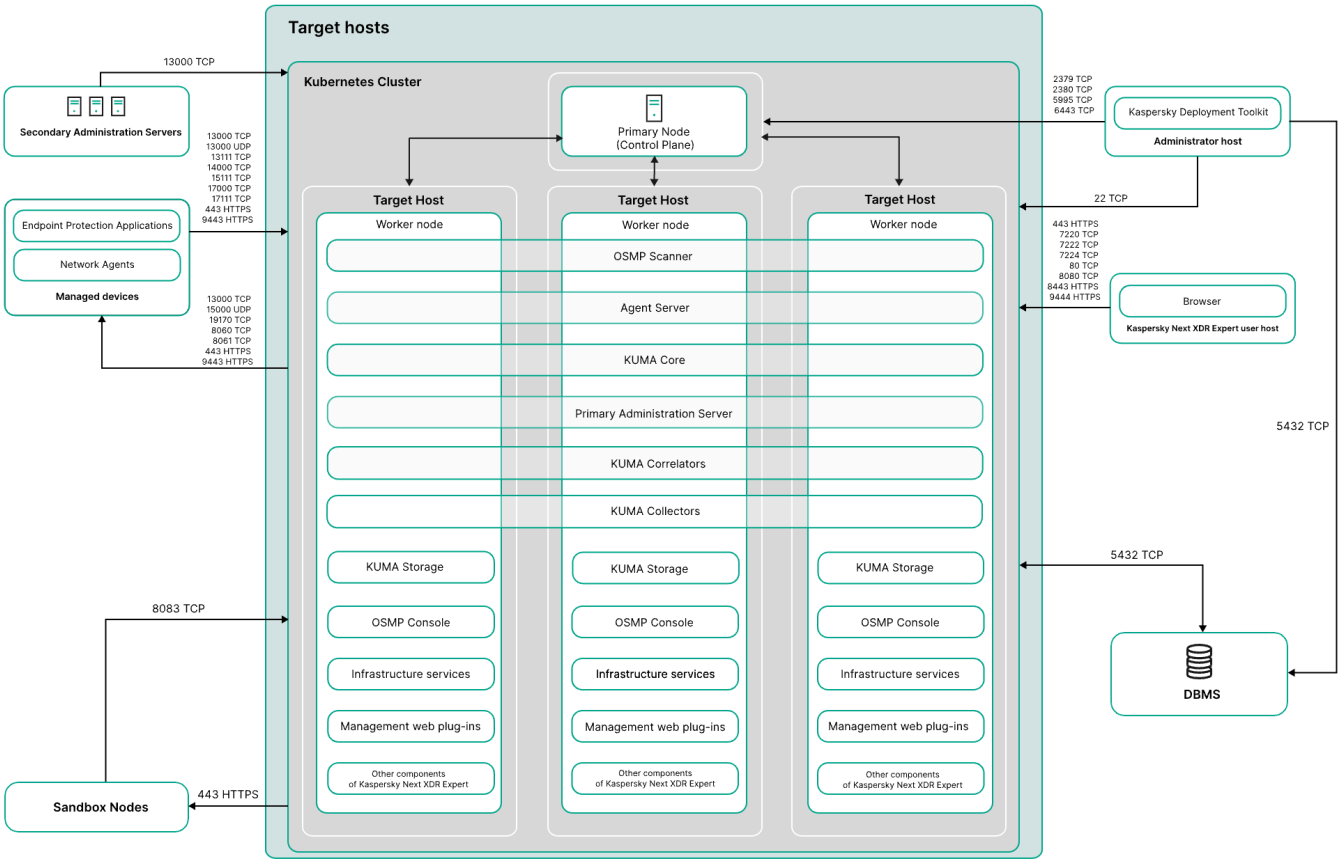
In a hierarchical deployment, multiple single nodes are deployed. Once all single nodes are deployed, one node is selected as a parent instance. Typically, the most powerful node becomes the parent, based on calculations. The remaining nodes are then connected to the parent instance.

To calculate the resources required for deployment, contact Technical Support.

Multi-node deployment scheme

The figure below shows the Kaspersky Next XDR Expert deployment scheme on multiple nodes.

On the scheme, the distribution of Kaspersky Next XDR Expert components among the worker nodes is shown as an example. Actual component distribution may vary.



Multi-node deployment scheme of Kaspersky Next XDR Expert

The multi-node deployment scheme of Kaspersky Next XDR Expert contains the following main components:

- **Administrator host**. On this host, an administrator uses Kaspersky Deployment Toolkit to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. The administrator host can be either located outside the Kubernetes cluster or located inside the cluster. If the administrator host is located inside the cluster, it will also act as the target host that is used for the operation of Kaspersky Next XDR Expert components.

- **DBMS server**. A server with an installed database management system is required for the proper function of Kaspersky Next XDR Expert components. The DBMS server is not included in the Kubernetes cluster. An administrator installs the DBMS manually on the separated server outside the Kubernetes cluster.

- **Kubernetes cluster**. A Kubernetes cluster includes the control plane node (also referred to as primary node during the deployment procedure) and, at a minimum, three worker nodes. The number of primary and worker nodes may vary. All Kaspersky Next XDR Expert components are deployed on worker nodes:

  - **Primary Administration Server**. The key component of endpoint protection of a client organization. Administration Server provides centralized deployment and management of endpoint protection through EPP applications, and allows you to monitor the endpoint protection status.

    This component is deployed on one of the worker nodes without node affinity. The Kubernetes scheduler selects a worker node based on available resources.

  - **OSMP Scanner**. Component that scans files by using Anti-Malware Engine, YARA, and Sandbox; sends files for analysis and generates reports.

    This component is deployed on one of the worker nodes without node affinity. The Kubernetes scheduler selects a worker node based on available resources.

  - **Agent server**. The component that is responsible for interactions between OSMP and Endpoint Agents.

    This component is deployed on one of the worker nodes without node affinity. The Kubernetes scheduler selects a worker node based on available resources.

  - **KUMA Core**. Central part of the SIEM component that serves as the foundation upon which all other services and components are built.

    This component is deployed on one of the worker nodes without node affinity. The Kubernetes scheduler selects a worker node based on available resources.

  - **OSMP Console**. The component that provides a web interface for OSMP.

    This component is deployed on all cluster nodes.

  - **Management web plug-ins**. Components used for remote administration of Kaspersky software by means of OSMP Console.

    These components are deployed on all cluster nodes.

  - **Infrastructure services and other components of Kaspersky Next XDR Expert**. Additional components that provide centralized asset management, data exchange, hierarchy building, authentication and authorization; as well as other functions, such as action auditing, security policy management, and update distribution. These are deployed on all cluster nodes.

- **KUMA services**. The KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on worker nodes, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster. The number of target hosts for KUMA services may vary. KUMA services are manually created by an administrator.

  On the scheme, KUMA services are installed entirely inside the cluster on worker nodes. In this case, KUMA collector and KUMA correlator are manually created by an administrator and, by default, are deployed without node affinity. Several KUMA storages are manually created on each node.

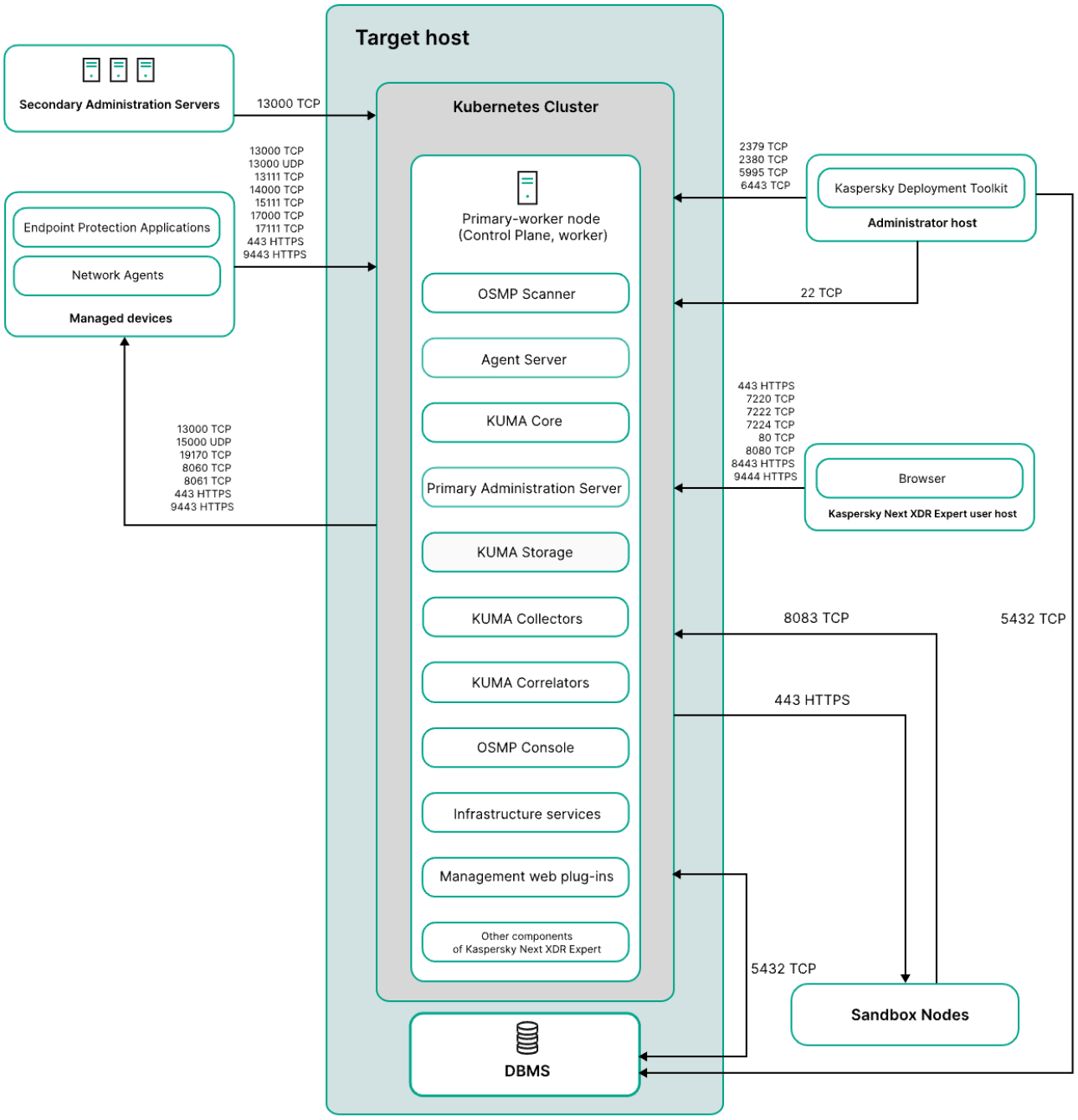
- **Kaspersky Next XDR Expert user host**. A user device that is used to sign in to OSMP Console.
- **Secondary Administration Servers** (optional). Secondary Administration Servers are used to create a Server hierarchy and for further integration with Endpoint Agents.
- **Managed devices**. Client devices protected by Kaspersky Next XDR Expert. Each managed device has Network Agent and an EPP application installed.
- **Sandbox nodes**. A Sandbox node, which can consist of one or more servers running Sandbox components, is a dedicated compute resource that hosts an isolated runtime environment and supporting services. This node is responsible for analyzing and processing files, links, URLs, and other objects entering the protected system. The Sandbox node operates independent of the Kubernetes cluster.

  The number of Sandbox servers may vary depending on the workload. An administrator installs Sandbox manually on each server.

This scheme does not provide all of the ports required for successful deployment. For the full list of ports, refer to Ports used by Kaspersky Next XDR Expert.

Single-node deployment scheme

The figure below shows the Kaspersky Next XDR Expert deployment scheme on a single node.



Single-node deployment scheme of Kaspersky Next XDR Expert

The single-node deployment scheme of Kaspersky Next XDR Expert contains the following main components:

- **Administrator host**. On this host, an administrator uses Kaspersky Deployment Toolkit to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. The administrator host can be either located outside the Kubernetes cluster or located inside the cluster. If the administrator host is located inside the cluster, it will also act as the target host that is used for the operation of Kaspersky Next XDR Expert components.

- **DBMS server**. A server with an installed database management system is required for the proper function of Kaspersky Next XDR Expert components. The DBMS server is not included in the Kubernetes cluster. An administrator installs the DBMS manually on the target host that will act as a primary worker node before the Kaspersky Next XDR Expert deployment.

- **Kubernetes cluster**. A Kubernetes cluster includes the target host that acts both as a control plane node (also referred to as primary node during the deployment procedure) and a worker node. All Kaspersky Next XDR Expert components are deployed on this primary worker node:

  - **Primary Administration Server**. The key component of endpoint protection of a client organization. Administration Server provides centralized deployment and management of endpoint protection through EPP applications, and allows you to monitor the endpoint protection status.

  - **OSMP Scanner**. Component that scans files by using Anti-Malware Engine, YARA, and Sandbox; sends files for analysis and generates reports.

  - **Agent server**. The component that is responsible for interactions between OSMP and Endpoint Agents.

  - **KUMA Core**. Central part of the SIEM component that serves as the foundation upon which all other services and components are built.

  - **OSMP Console**. The component that provides a web interface for OSMP.

  - **Management web plug-ins**. Components used for remote administration of Kaspersky software by means of OSMP Console.

  - **Infrastructure services and other components of Kaspersky Next XDR Expert**. Additional components that provide centralized asset management, data exchange, hierarchy building, authentication and authorization; as well as other functions, such as action auditing, security policy management, and update distribution.

- **KUMA services**. The KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on worker nodes, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster. The number of target hosts for KUMA services may vary. KUMA services are manually created by an administrator. On the scheme, KUMA services are installed entirely inside the cluster on worker nodes.

- **Kaspersky Next XDR Expert user host**. A user device that is used to sign in to OSMP Console.

- **Secondary Administration Servers** (optional). Secondary Administration Servers are used to create a Server hierarchy and for further integration with Endpoint Agents.

- **Managed devices**. Client devices protected by Kaspersky Next XDR Expert. Each managed device has Network Agent and an EPP application installed.

- **Sandbox nodes**. A Sandbox node, which can consist of one or more servers running Sandbox components, is a dedicated compute resource that hosts an isolated runtime environment and supporting services. This node is responsible for analyzing and processing files, links, URLs, and other objects entering the protected system. The Sandbox node operates independent of the Kubernetes cluster.

  The number of Sandbox servers may vary depending on the workload. An administrator installs Sandbox manually on each server.

This scheme does not provide all of the ports required for successful deployment. For the full list of ports, refer to Ports used by Kaspersky Next XDR Expert.

Ports used by Kaspersky Next XDR Expert

For correct interaction between the administrator host and target hosts, you must provide connection access from the administrator host to the target hosts by the ports listed in the table below. These ports cannot be changed.

For interaction between the administrator host and hosts that are used for the installation of the KUMA services and are located outside the Kubernetes cluster, you must provide access only by TCP 22 port.

Ports used for interaction between the administrator host and target hosts

Port	Protocol	Direction	Source	Destination	Port purpose
22	TCP	Inbound	Administrator host	Target hosts	Providing the SSH connection from the administrator host to the target hosts. Providing the SSH connection from the administrator host to the hosts that are used for the installation of the external KUMA services.
2379, 2380	TCP	Inbound	Administrator host	Target host (primary node)	Connection to the Kubernetes storage (ETCD).
5995	TCP	Inbound	Administrator host	Target host (primary node)	Connection to the Docker registry.
6443	TCP	Inbound	Administrator host	Target host (primary node)	Connection to the Kubernetes API.

For properly work of the Kaspersky Next XDR Expert components, the target hosts must be located in the same broadcast domain.

The table below contains the ports that must be opened on the firewalls of all target hosts of the cluster. These ports cannot be changed.

If you use the firewalld or UFW firewall on your target hosts, KDT opens the required ports on the firewalls automatically. Otherwise, you can open the listed ports manually before you deploy Kaspersky Next XDR Expert.

Required ports used by the Kaspersky Next XDR Expert components

Port	Protocol	Direction	Source	Destination	Port purpose
80, 8080	TCP (HTTP)	Inbound	Browser	Target hosts of the cluster	Receiving connections from browser. Redirecting to the 443 TCP (HTTPS) port.
443, 8443	TCP (HTTPS)	Inbound	Browser	Target hosts of the cluster	Receiving connections from browser. Receiving connections to the Administration Server over OpenAPI. Used to automate scenarios for working with the Administration Server.
13000	TCP	Inbound	Network Agent or secondary Administration Server	Target hosts of the cluster	Receiving connections from Network Agents and secondary Administration Servers.
13000	UDP	Inbound	Network Agent	Target hosts of the cluster	Receiving information about devices that were turned off from Network Agents.

Port	Protocol	Direction	Source	Destination	Port purpose
14000	TCP	Inbound	Network Agent	Target hosts of the cluster	Receiving connections from Network Agents.
17000	TCP	Inbound	Managed devices	Target hosts of the cluster	Receiving connections for application activation from managed devices (except for mobile devices).
19170	TCP (HTTPS)	Outbound	Target hosts of the cluster	Managed devices	Remote access to managed devices by using OSMP Console.
7210	TCP	Inbound	KUMA target hosts	Target hosts of the cluster	Receiving of the KUMA configuration from the KUMA Core server.
7220	TCP	Inbound	Browser	Target hosts of the cluster	Receiving connections from browser.
7222	TCP	Inbound	Browser	Target hosts of the cluster	Reversing proxy in the CyberTrace system.
7224	TCP	Inbound	Browser	Target hosts of the cluster	Callbacks for Identity and Access Manager (IAM).
8429	TCP	Inbound	KUMA agent	Target hosts of the cluster	Logging KUMA agent performance metrics.

The table below contains the ports that are not opened by default on the firewalls during the Kaspersky Next XDR Expert deployment. These ports cannot be changed.

If you need to perform actions listed in the **Port purpose** column of the table below, you can open the corresponding ports on the firewalls of all target hosts manually.

Optional ports on the firewall used by the Kaspersky Next XDR Expert components

Port	Protocol	Direction	Source	Destination	Port purpose
8060	TCP	Outbound	Target hosts of the cluster	Managed devices	Transmitting published installation packages to managed devices.
8061	TCP	Outbound	Target hosts of the cluster	Managed devices	Transmitting published installation packages to managed devices.
13111	TCP	Inbound	Managed devices	Target hosts of the cluster	Receiving requests from managed devices to KSN proxy server.
15111	UDP	Inbound	Managed devices	Target hosts of the cluster	Receiving requests from managed devices to KSN proxy server.
17111	TCP	Inbound	Managed devices	Target hosts of the cluster	Receiving requests from managed devices to KSN proxy server.
5432	TCP	Inbound	DBMS (PostgreSQL)	Target hosts of the cluster	Interaction with the DBMS (PostgreSQL). This port is used only if you perform the demonstration deployment and install the DBMS on the target host inside the Kubernetes cluster.

The table below contains the ports that must be opened for functioning of the Kubernetes cluster and infrastructure components. These ports cannot be changed.

If you use the firewalld or UFW firewall on your target hosts, the KDT opens the required ports on the firewalls automatically. Otherwise, you can open the listed ports manually before you deploy Kaspersky Next XDR Expert.

Ports used by the Kubernetes cluster and infrastructure components

Port	Protocol	Node
80	TCP	Primary node
443	TCP	Primary node
10250	TCP	Primary node
9443	TCP	Primary node
6443	TCP	Primary node
8132	TCP	Primary node

Port	Protocol	Node
5995	TCP	Primary node
80	TCP	Worker node
443	TCP	Worker node
179	TCP	Worker node
10250	TCP	Worker node
10255	TCP	Worker node
9443	TCP	Worker node
6443	TCP	Worker node
9500	TCP	Worker node
9501	TCP	Worker node
9502	TCP	Worker node
9503	TCP	Worker node
8500	TCP	Worker node
8501	TCP	Worker node
3260	TCP	Worker node
8000	TCP	Worker node
8002	TCP	Worker node
2049	TCP	Worker node
3370	TCP	Worker node
179	UDP	Worker node
51820	UDP	Worker node
51821	UDP	Worker node

For correct work of the KUMA services that are not included in a Kubernetes cluster, you must open the ports listed in the table below. The table below shows the default network ports values. These ports automatically open during the KUMA installation.

Ports used for the interaction with the external KUMA services

Port	Protocol	Direction	Source	Destination	Port purpose
8123	HTTPS	Inbound	Storage service	ClickHouse cluster node	Writing and receiving normalized events in the ClickHouse cluster.
9009	HTTPS	Inbound/Outbound	ClickHouse cluster replica	ClickHouse cluster replica	Internal communication between ClickHouse cluster replicas for transferring data of the cluster.
2181	TCP	Inbound	ClickHouse cluster nodes	ClickHouse keeper replication coordination service	Receiving and writing of replication metadata by replicas of ClickHouse servers.
2182	TCP	Inbound/Outbound	ClickHouse keeper replication coordination service	ClickHouse keeper replication coordination service	Internal communication between replication coordination services to reach a quorum.
8001	TCP	Inbound	Victoria Metrics	ClickHouse server	Receiving ClickHouse server operation metrics.
9000	TCP	Inbound	ClickHouse client	ClickHouse cluster node	Writing and receiving data in the ClickHouse cluster.

If you create an additional KUMA service (collector, correlator or storage) on a server, you need to manually open a port that corresponds to the created service on the server. You can use port TCP 7221 or other port used for service installation.

If the out of the box example services are used, the following ports automatically open during the Kaspersky Next XDR Expert deployment:

- 7230 TCP
- 7231 TCP
- 7232 TCP
- 7233 TCP
- 7234 TCP
- 7235 TCP
- 7236 TCP
- 5140 TCP
- 5140 UDP
- 5141 TCP
- 5144 UDP
- 5150 UDP

Preparation work and deployment

This section describes how to prepare the infrastructure for the Kaspersky Next XDR Expert deployment, set the installation parameters that are specific for the multi-node or single-node deployment, as well as how to use the Configuration wizard to generate the configuration file.

You will find out how to install Kaspersky Next XDR Expert according to the multi-node and single-node deployment schemes. Also, this section contains information on how to deploy multiple Kubernetes clusters with Kaspersky Next XDR Expert instances and switch between them by using KDT.

Multi-node deployment: Preparing the administrator and target hosts

Preparing for a multi-node deployment includes configuring the administrator and target hosts. After preparing hosts and specifying the configuration file, you will be able to deploy Kaspersky Next XDR Expert on target hosts by using KDT.

Preparing the administrator host

You first need to prepare a device that will act as the administrator host from which KDT will launch. This host can be either included in the Kubernetes cluster that is created by KDT during the deployment, or not. If the administrator host is not included in the cluster, it will be used only to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. If the administrator host is included in the cluster, it will also act as a target host that is used for the operation of Kaspersky Next XDR Expert components.

To prepare the administrator host:

1. Make sure that the hardware and software on the administrator host meet the requirements.

   If the administrator host is not included in the Kubernetes cluster, use the KDT requirements.

   If the administrator host is included in the cluster, use the hardware requirements for the multi-node deployment and the software requirements for OSMP.

2. Allocate at least 15 GB of free space in the temporary files directory (by default, `/tmp`) for KDT. If you do not have enough free space in this directory, run the following command to specify the path to another directory:

   ```
   export TMPDIR=<new_directory>/tmp
   ```

3. Install the package for Docker version 23 or later. Do not install unofficial distributions of Docker packages from the operating system maintainer repositories.

   The disk partition for Docker data storage (by default, `/var/lib/docker`) must have at least 60 GB of available space.

4. Perform the post-installation steps to configure the administration host for proper functioning with Docker:

   ```
   sudo groupadd docker
   sudo usermod -aG docker $USER
   newgrp docker
   sudo systemctl enable docker.service
   sudo systemctl enable containerd.service
   sudo systemctl status docker.service
   ```

   Make sure the Docker service is in the **Active** status.

   If a key cannot be downloaded automatically, download it manually:

   a. Open the https://download.docker.com/linux/ubuntu/gpg link in your browser save the key as docker.asc.

   b. Run the following command:

   ```
   sudo mv docker.asc /etc/apt/keyrings/docker.asc
   ```

5. For the administrator host that will be included in the cluster, perform additional preparatory steps:

   a. Since the device will act as the administrator and target host, make sure that it meets the requirements for multi-node deployment.

   b. Make sure that the cgroup v2 technology is supported on the administrator host.

   a. Since the device will act as the administrator and target host, make sure that it meets the requirements for the single-node deployment (because the demonstration and single-node deployment have a similar deployment scheme).

   b. Make sure that the cgroup v2 technology is supported on the administrator host.

      The cgroup v2 technology is supported for the Linux kernel version 2.6.24 or later.

   c. Install the uidmap package on the administrator host.

      Check that the /etc/subgid and /etc/subuid files contain the user account under which KDT will be launched. To do this, you can run the following command:

      ```
      getsubids USER
      ```

      If this command does not return a result, you must add the user account to the /etc/subgid and /etc/subuid files manually in the following format:

      ```
      <username>:<min_subid>:<range_length>
      ```

      where

      - `<username>`—Username of the account under which KDT will be launched.
      - `<min_subid>`—Minimum subuid value.
      - `<range_length>`—Number of subuids allocated for the user `<username>`.

Preparing the target hosts

The target hosts are physical or virtual machines that are used to deploy Kaspersky Next XDR Expert and included in the Kubernetes cluster. Kaspersky Next XDR Expert components work on these hosts.

> One of the target hosts can be used as the administrator host. In this case, you must prepare this host as the administrator host, as described in the previous procedure, and then prepare the target host.

A minimum cluster configuration for the multi-node deployment includes four nodes:

- One primary node

  The primary node is intended for managing the cluster, storing metadata, and distributing the workload.

- Three worker nodes

  The worker nodes are intended for performing the workload of the Kaspersky Next XDR Expert components.

  The number of primary and worker nodes in the cluster may vary. For optimal workload distribution between nodes, it is recommended to use nodes with approximately the same performance.

KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on worker nodes, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster.

If you have multiple primary nodes, a load balancer is required to distribute incoming requests across the primary nodes and ensure an even workload. If you want to use an external load balancer, allocate a separate host outside the Kubernetes cluster and specify its parameters in the configuration file. If you do not want to use an external load balancer, a worker node will act as the load balancer.

To prepare the target hosts:

1. Make sure that the hardware and software on the target hosts meet the requirements for multi-node deployment, and the target hosts are located in the same broadcast domain.

   If KUMA services are installed inside the cluster, target hosts must meet the hardware, software, and installation requirements that are necessary for installing the KUMA services.

   For proper functioning of Kaspersky Next XDR Expert, the Linux kernel version must be 5.15.0.107 or later on the target hosts with the Ubuntu family of operating systems.

   Docker must not be installed on the target hosts, except the target host that will be used as the administrator host. KDT will install all necessary software and dependencies during the deployment.

2. Ensure that the filesystem mounted at `/var` has at least 700 GB of free space on each target host.

3. On each target host, install the sudo package, if this package is not already installed. For the Debian family of operating systems, install the UFW package on the target hosts.

4. On each target host, configure the /etc/environment file. If your organization's infrastructure uses a proxy server to access the internet, configure the internet access by using the proxy server on the target hosts.

5. On the primary nodes with the UFW configuration, allow IP forwarding. In the `/etc/default/ufw` file, set `DEFAULT_FORWARD_POLICY` to `ACCEPT`.

6. Provide access to the package repository that stores the following packages required for Kaspersky Next XDR Expert:

   - nfs-common
   - containerd
   - cron
   - cryptsetup
   - cryptsetup-bin
   - netcat-traditional
   - nfs-common
   - open-iscsi
   - python3-cffi-backend

- python3-cryptography
- python3-netaddr
- python-apt-common
- systemd
- tar

KDT will try to install these packages during the deployment from the package repository. You can also install these packages manually.

7. Ensure that the following packages are installed:

   - On each target host: python3-apt
   - On primary nodes: curl

   These packages are not installed during the deployment from the package repository by using KDT. You must install these packages manually, if they are not already installed.

8. Disable ARP spoofing protection on the hypervisor or network infrastructure for the target hosts.

9. Reserve static IP addresses for the target hosts and the Kubernetes cluster gateway.

   The Kubernetes cluster gateway is intended for connecting to the Kaspersky Next XDR Expert components installed inside the Kubernetes cluster. The gateway IP address is an IPv4 address (for example, 10.80.23.182). It is specified in the configuration file (the `ingress_ip` parameter).

   Make sure that the target hosts, the Kubernetes cluster gateway, and the DBMS host are located in the same broadcast domain.

   > The Kubernetes cluster gateway and target hosts require IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on the target hosts and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks.

10. On your DNS server, register the service FQDNs to connect to the Kaspersky Next XDR Expert services.

    By default, the Kaspersky Next XDR Expert services are available at the following addresses:

    - <console_host>.<smp_domain>—Access to the OSMP Console interface.
    - <ksc_host>.<smp_domain>—Interaction with Administration Server.
    - <kuma_host>.<smp_domain>—Access to the SIEM component.
    - <api_host>.<smp_domain>—Access to the Kaspersky Next XDR Expert API.
    - <monitoring_host>.<smp_domain>—Access to OSMP metrics.

      Where <console_host>, <ksc_host>, <kuma_host>, <api_host>, and <monitoring_host> are service host names, <smp_domain> is a service domain name. These parameters are parts of the service FQDNs, which you can specify in the configuration file. If you do not specify custom values of service host names, the default values are used: `console_host`—`"console"`, `ksc_host`—`"admsrv"`, `kuma_host`—`"kuma"`, `api_host`—`"api"`, `monitoring_host`—`"monitoring"`.

The listed service FQDNs must be resolved to the IP address of the Kubernetes cluster. For example, if the Kubernetes cluster address is 10.80.23.182, service FQDNs are resolved to the IP address as follows:

- <console_host>.<smp_domain>—10.80.23.182
- <ksc_host>.<smp_domain>—10.80.23.182
- <kuma_host>.<smp_domain>—10.80.23.182
- <api_host>.<smp_domain>—10.80.23.182
- <monitoring_host>.<smp_domain>—10.80.23.182

11. On each target host, create the accounts that will be used for the Kaspersky Next XDR Expert deployment.

    These accounts are used for the SSH connection and must be able to elevate privileges (sudo) without entering a password. To do this, add the created user accounts to the `/etc/sudoers` file.

    View details

    To add the created user accounts to the `/etc/sudoers` file:

    1. Run the following commands:

       ```
       sudo useradd -m -s /bin/bash <user_name>
       sudo usermod -aG sudo <user_name>
       ```

    2. Open the `/etc/sudoers` file by using the following command:

       ```
       sudo visudo
       ```

    3. Add the following line to the end of the section `# Allow members of group sudo to execute any command`:

       ```
       <user_name> ALL=(ALL) NOPASSWD: ALL
       ```

       where `<user_name>` is the name of the created user.

    4. Save and close the `/etc/sudoers` file.

12. Configure the SSH connection between the administrator and target hosts:

    a. On the administrator host, generate SSH keys by using the ssh-keygen utility without a passphrase.

    b. Copy the public key to every target host and, if a load balancer is used, to the load balancer host (for example, to the `/home/<user_name>/.ssh` directory) by using the ssh-copy-id utility.

       If you use a target host as the administrator host, you must copy the public key to it, too.

13. For proper function of the Kaspersky Next XDR Expert components, provide network access between the target hosts and open the required ports on the firewall of the administrator and target hosts, if necessary.

14. Configure time synchronization over Network Time Protocol (NTP) on the administrator and target hosts.

15. If necessary, prepare custom certificates for working with Kaspersky Next XDR Expert public services.

    You can use one intermediate certificate that is issued off the organization's root certificate or leaf certificates for each of the services. The prepared custom certificates will be used instead of self-signed certificates.

Single-node deployment: Preparing the administrator and target hosts

Preparing for a single-node deployment includes configuring the administrator and target hosts. In the single-node configuration, the Kubernetes cluster and Kaspersky Next XDR Expert components are installed on one target host. After preparing the target host and specifying the configuration file, you will be able to deploy Kaspersky Next XDR Expert on the target host by using KDT.

Preparing the administrator host

You first need to prepare a device that will act as the administrator host from which KDT will launch. This host can be either included in the Kubernetes cluster that is created by KDT during the deployment, or not. If the administrator host is not included in the cluster, it will be used only to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. If the administrator host is included in the cluster, it will also act as a target host that is used for operation of Kaspersky Next XDR Expert components. In this case, only one host will be used for deployment and operation of the solution.

To prepare the administrator host:

1. Make sure that the hardware and software on the administrator host meet the requirements.

   If the administrator host is not included in the Kubernetes cluster, use the KDT requirements.

   If the administrator host is included in the cluster, use the hardware requirements for the single-node deployment and the software requirements for OSMP.

2. Allocate at least 15 GB of free space in the temporary files directory (by default, `/tmp`) for KDT. If you do not have enough free space in this directory, run the following command to specify the path to another directory:

   ```
   export TMPDIR=<new_directory>/tmp
   ```

3. Install the package for Docker version 23 or later, and then perform the post-installation steps to configure the administration host for proper functioning with Docker.

   The disk partition for Docker data storage (by default, `/var/lib/docker`) must have at least 60 GB of available space.

   Do not install unofficial distributions of Docker packages from the operating system maintainer repositories.

4. For the administrator host that will be included in the cluster, perform additional preparatory steps:

   a. Since the device will act as the administrator and target host, make sure that it meets the requirements for the single-node deployment.

   b. Make sure that the cgroup v2 technology is supported on the administrator host.

   a. Since the device will act as the administrator and target host, make sure that it meets the requirements for the single-node deployment (because the demonstration and single-node deployment have a similar deployment scheme).

   b. Make sure that the cgroup v2 technology is supported on the administrator host.

      The cgroup v2 technology is supported for the Linux kernel version 2.6.24 or later.

   c. Install the uidmap package on the administrator host.

      Check that the /etc/subgid and /etc/subuid files contain the user account under which KDT will be launched. To do this, you can run the following command:

      ```
      getsubids USER
      ```

      If this command does not return a result, you must add the user account to the /etc/subgid and /etc/subuid files manually in the following format:

      ```
      <username>:<min_subid>:<range_length>
      ```

      where

      - `<username>`—Username of the account under which KDT will be launched.
      - `<min_subid>`—Minimum subuid value.
      - `<range_length>`—Number of subuids allocated for the user `<username>`.

Preparing the target host

The target host is a physical or virtual machine that is used to deploy Kaspersky Next XDR Expert and included in the Kubernetes cluster. The target host manages the Kubernetes cluster and the Kaspersky Next XDR Expert components that work on this host, and stores metadata.

A minimum cluster configuration for the single-node deployment includes one target host, which acts as the primary and worker nodes. On this primary worker node, the Kubernetes cluster and Kaspersky Next XDR Expert components are installed.

KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on the primary worker node, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster.

If you want to run the Kaspersky Next XDR Expert deployment from the target host, you must prepare this host as the administrator host, as described in the previous procedure, and then prepare the target host.

To prepare the target host:

1. Make sure that the hardware and software on the target host meet the requirements for the single-node deployment.

If KUMA services are installed inside the cluster, the target host must meet the hardware, software, and installation requirements for installing the KUMA services.

Since the DBMS will be installed on the target host, you also need to make sure that the target host meets the requirements for the DBMS.

> For proper functioning of Kaspersky Next XDR Expert, the Linux kernel version must be 5.15.0.107 or later on the target host with the Ubuntu family of operating systems.

> Do not install Docker on the target host unless the target host will be used as the administrator host. KDT will install all necessary software and dependencies during the deployment.

2. Ensure that the filesystem mounted at `/var` has at least 360 GB of free space on the target host.

3. Install the DBMS on the target host.

   You must install the DBMS before the Kaspersky Next XDR Expert deployment, because in this case the DBMS and Kaspersky Next XDR Expert components will be installed on the same target host, and the DBMS will not be included in the Kubernetes cluster.

4. Install the sudo package, if this package is not already installed. For the Debian family of operating systems, install the UFW package.

5. Configure the /etc/environment file. If your organization's infrastructure uses a proxy server to access the internet, configure the internet access by using the proxy server on the target hosts.

6. If the primary worker node has the UFW configuration, allow IP forwarding. In the `/etc/default/ufw` file, set `DEFAULT_FORWARD_POLICY` to `ACCEPT`.

7. Provide access to the package repository that stores the following packages required for Kaspersky Next XDR Expert:

   - nfs-common
   - containerd
   - cron
   - cryptsetup
   - cryptsetup-bin
   - netcat-traditional
   - nfs-common
   - open-iscsi
   - python3-cffi-backend
   - python3-cryptography
   - python3-netaddr

- python-apt-common
- systemd
- tar

KDT will try to install these packages during the deployment from the package repository. You can also install these packages manually.

8. Ensure that the following packages are installed on the primary worker node:

   - python3-apt
   - curl

   These packages are not installed during the deployment from the package repository by using KDT. You must install these packages manually, if they are not already installed.

9. Disable ARP spoofing protection on the hypervisor or network infrastructure for the target hosts.

10. Reserve static IP addresses for the target host and for the Kubernetes cluster gateway.

    The Kubernetes cluster gateway is intended for connecting to the Kaspersky Next XDR Expert components installed inside the Kubernetes cluster. The gateway IP address is an IPv4 address (for example, 10.80.23.182). It is specified in the configuration file (the `ingress_ip` parameter).

    Make sure that the target host and the Kubernetes cluster gateway are located in the same broadcast domain.

    > The Kubernetes cluster gateway and target hosts require IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on the target hosts and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks.

11. On your DNS server, register the service FQDNs to connect to the Kaspersky Next XDR Expert services.

    By default, the Kaspersky Next XDR Expert services are available at the following addresses:

    - <console_host>.<smp_domain>—Access to the OSMP Console interface.
    - <ksc_host>.<smp_domain>—Interaction with Administration Server.
    - <kuma_host>.<smp_domain>—Access to the SIEM component.
    - <api_host>.<smp_domain>—Access to the Kaspersky Next XDR Expert API.
    - <monitoring_host>.<smp_domain>—Access to OSMP metrics.

      Where <console_host>, <ksc_host>, <kuma_host>, and <api_host>, and <monitoring_host> are service host names, <smp_domain> is a service domain name. These parameters are parts of the service FQDNs, which you can specify in the configuration file. If you do not specify custom values of service host names, the default values are used: `console_host—"console"`, `ksc_host—"admsrv"`, `kuma_host—"kuma"`, `api_host—"api"`, `monitoring_host—"monitoring"`.

The listed service FQDNs must be resolved to the IP address of the Kubernetes cluster. For example, if the Kubernetes cluster address is 10.80.23.182, service FQDNs are resolved to the IP address as follows:

- <console_host>.<smp_domain>—10.80.23.182
- <ksc_host>.<smp_domain>—10.80.23.182
- <kuma_host>.<smp_domain>—10.80.23.182
- <api_host>.<smp_domain>—10.80.23.182
- <monitoring_host>.<smp_domain>—10.80.23.182

12. Create the user accounts that will be used for the Kaspersky Next XDR Expert deployment.

    These accounts are used for the SSH connection and must be able to elevate privileges (sudo) without entering a password. To do this, add the created user accounts to the `/etc/sudoers` file.

    View details

    To add the created user accounts to the `/etc/sudoers` file:

    1. Run the following commands:

       ```
       sudo useradd -m -s /bin/bash <user_name>
       sudo usermod -aG sudo <user_name>
       ```

    2. Open the `/etc/sudoers` file by using the following command:

       ```
       sudo visudo
       ```

    3. Add the following line to the end of the section `# Allow members of group sudo to execute any command`:

       ```
       <user_name> ALL=(ALL) NOPASSWD: ALL
       ```

       where `<user_name>` is the name of the created user.

    4. Save and close the `/etc/sudoers` file.

13. Configure the SSH connection between the administrator and target hosts:

    a. On the administrator host, generate SSH keys by using the ssh-keygen utility without a passphrase.

    b. Copy the public key to the target host (for example, to the `/home/<user_name>/.ssh` directory) by using the ssh-copy-id utility.

       If you use the target host as the administrator host, you must copy the public key to it, too.

14. For proper function of the Kaspersky Next XDR Expert components, open the required ports on the firewall of the administrator and target hosts, if necessary.

15. Configure time synchronization over Network Time Protocol (NTP) on the administrator and target hosts.

16. If necessary, prepare custom certificates for working with Kaspersky Next XDR Expert public services.

    You can use one intermediate certificate that is issued off the organization's root certificate or leaf certificates for each of the services. The prepared custom certificates will be used instead of self-signed certificates.

KUMA installation requirements

General application installation requirements

You can install the application on the following operating systems:

- Oracle Linux
- Astra Linux
- Ubuntu

  Supported operating system versions are listed in the Hardware and software requirements section.

  Supported configurations are Server and Server with GUI support. In the Server with GUI support configuration, you do not need to install additional packages for reporting.

Before deploying the application, make sure the following conditions are met:

- Servers on which you want to install the components satisfy the hardware and software requirements.
- Ports to be used by the installed instance of KUMA are available.
- KUMA components are addressed using the fully qualified domain name (FQDN) of the host in the hostname.example.com format. Before you install the application, make sure that the correct host FQDN is returned when you run the following command:

  ```
  hostname -f
  ```

- The name of server on which you are running the installer is not `localhost` or `localhost.<domain>`.
- The name of the server on which you are installing KUMA Core does not start with a numeral.
- Time synchronization over Network Time Protocol (NTP) is configured on all servers with KUMA services.

Installation requirements for Oracle Linux, Astra Linux, Ubuntu 22.04 LTS operating systems

	Oracle Linux	Astra Linux	Ubuntu 22.04 LTS
Python version	From 3.6 to 3.11. Versions 3.12 and later are not supported.	From 3.6 to 3.11. Versions 3.12 and later are not supported.	From 3.6 to 3.11. Versions 3.12 and later are not supported.
SELinux module	Disabled	Disabled	Disabled

	Oracle Linux	Astra Linux	Ubuntu 22.04 LTS
Package manager	pip3	pip3	pip3

Basic packages	• netaddr
	• firewalld
	• compat-openssl11
	You need to install this package on the Oracle Linux 9 host on which the KUMA Core will be deployed outside the cluster.
	More on upgrading from Oracle Linux 8.x to Oracle Linux 9.x.
	To install the packages, run the following commands:
	`pip3 install netaddr`
	`yum install firewalld`
	`yum install compat-openssl11`
		• python3-apt
		• curl
		• libcurl4
		To install the packages, run the following command:
		`apt install python3-apt curl libcurl4`
			• python3-apt
			• curl
			• libcurl4
			• openssl 1.1.1
			• acl
			To install the packages, run the following command:
			`apt install python3-apt curl libcurl4 acl`
			You can download the openssl1.1.1 package from the official website of Ubuntu and install it using the following command:
			`dpkg -i libssl1.1_1.1.1f-1ubuntu2_amd64.deb`

Dependent packages	No value.	• netaddr • python3-cffi-backend To install the packages, run the following command: `apt install python3-netaddr python3-cffi-backend` If you plan to query Oracle DB databases from KUMA, you need to install the Astra Linux libaio1 package.	• netaddr • python3-cffi-backend To install the packages, run the following command: `apt install python3-netaddr python3-cffi-backend`
Packages that must be installed on a device with the KUMA Core for correct generation and downloading of reports	• nss • gtk2 • atk • libnss3.so • libatk-1.0.so.0 • libxkbcommon • libdrm • at-spi2-atk • mesa-libgbm • alsa-lib • cups-libs • libXcomposite • libXdamage • libXrandr To install the packages, run the following command: `yum install nss gtk2 atk libnss3.so libatk-1.0.so.0 libxkbcommon libdrm at-spi2-atk mesa-libgbm alsa-lib cups-libs libXcomposite libXdamage libXrandr`	• libgtk2.0.0 • libnss3 • libatk-adaptor • libdrm-common • libgbm1 • libxkbcommon0 • libasound2 To install the packages, run the following command: `apt install libgtk2.0.0 libnss3 libatk-adaptor libdrm-common libgbm1 libxkbcommon0 libasound2`	• libatk1.0-0 • libgtk2.0-0 • libatk-bridge2.0-0 • libcups2 • libxcomposite-dev • libxdamage1 • libxrandr2 • libgbm-dev • libxkbcommon-x11-0 • libpangocairo-1.0-0 • libasound2 To install the packages, run the following command: `apt install libatk1.0-0 libgtk2.0-0 libatk-bridge2.0-0 libcups2 libxcomposite-dev libxdamage1 libxrandr2 libgbm-dev libxkbcommon-x11-0 libpangocairo-1.0-0 libasound2`
User permissions required to install the application	No value.	You need to assign the required permissions to the user that will be installing the application: `sudo pdpl-user -i 63 < user that will be installing the application >`	No value.

Preparing the hosts for installation of the KUMA services

The KUMA services (collectors, correlators, and storages) are installed on the KUMA target hosts that are located outside the Kubernetes cluster.

Access to KUMA services is performed by using the KUMA target host FQDNs. The administrator host must be able to access the KUMA target hosts by its FQDNs.

If you plan to use multiple tenants in your infrastructure, create a separate collector for each tenant.

To prepare the KUMA target hosts for installation of the KUMA services:

1. Ensure that the hardware, software, and installation requirements are met.

2. Specify the host names.

   You must specify the FQDN, for example: kuma1.example.com.

   We do not recommend changing the KUMA host name after installation. This will make it impossible to verify the authenticity of certificates and will disrupt the network communication between the application components.

3. Run the following commands:

   ```
   hostname -f
   hostnamectl status
   ```

   Compare the output of the `hostname -f` command and the value of the `Static hostname` field in the `hostnamectl status` command output. These values must match the FQDN of the device.

4. Configure the SSH connection between the administrator host and KUMA target hosts.

   Use the SSH keys created for the target hosts. Copy the public key to the KUMA target hosts by using the ssh-copy-id utility.

5. Register the KUMA target hosts in your organization's DNS zone, to allow host names to be translated to IP addresses.

6. Ensure time synchronization over Network Time Protocol (NTP) is configured on all KUMA target hosts. The KUMA services (collectors, correlators, and storages) are installed on the KUMA target hosts that are located outside the Kubernetes cluster.

The hosts are ready for installation of the KUMA services.

Installing a database management system

Kaspersky Next XDR Expert supports PostgreSQL or Postgres Pro database management systems (DBMS). For the full list of supported DBMSs, refer to the Hardware and software requirements.

Each of the following Kaspersky Next XDR Expert components requires a database:

- Administration Server
- Automation Platform
- Incident Response Platform (IRP)
- Identity and Access Manager (IAM)

Each of the components must have a separate database within the same instance of DBMS. The DBMS instance must be installed outside the Kubernetes cluster on a separate server. We recommend that you install the DBMS instance outside the Kubernetes cluster.

For the DBMS installation, KDT requires a privileged DBMS account that has permissions to create databases and other DBMS accounts. KDT uses the default DBMS account with superuser permissions, or another DBMS account with the same permissions, to create the databases and other DBMS accounts required for the Kaspersky Next XDR Expert components.

For the multi-node deployment, you must manually install the DBMS on the separated server that will not be included the Kubernetes cluster.

For the single-node deployment, you must manually install the DBMS on the target host before the Kaspersky Next XDR Expert deployment. In this case, the DBMS and Kaspersky Next XDR Expert components are installed on the same target host, but the DBMS is not included in the Kubernetes cluster.

For information about how to install the selected DBMS, refer to its documentation.

PostgreSQL installation command example

To install PostgreSQL, run the following commands:

```
sudo apt install -y postgresql-<version>
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

Where `<version>` is the supported PostgreSQL version.

After you install the DBMS, you need to configure the DBMS server parameters to ensure proper operation of the DBMS with Open Single Management Platform.

Configuring the PostgreSQL or Postgres Pro server for working with Open Single Management Platform

To successfully configure PostgreSQL or Postgres Pro server for working with Administration Server, you must use the default server parameters except for the server parameters specified in this article.

Kaspersky Next XDR Expert supports PostgreSQL or Postgres Pro database management systems (DBMS). For the full list of supported DBMSs, refer to the Hardware and software requirements. Consider configuring the DBMS server parameters to optimize the DBMS work with Administration Server. Compliance of the DBMS server parameters with the recommended values is one of the prerequisites for installing Kaspersky Next XDR Expert, which is verified by KDT.

Find the DBMS configuration file:

- The default path to the PostgreSQL configuration file is: `/etc/postgresql/<VERSION>/main/postgresql.conf`
- The default path to the Postgres Pro configuration file is: `/var/lib/pgpro/std-<VERSION>/data/postgresql.conf`

If there is no configuration file at the default path, run the following commands to find the correct path:

```
sudo su - postgres
psql -c 'SHOW config_file'
```

The command response will contain the path to the configuration file.

Then log out from the `postgres` user by running the following command:

```
exit
```

Recommended parameters for PostgreSQL and Postgres Pro DBMS for work with Administration Server:

- `listen_addresses = '*'`

  Specifies IP interfaces that Administration Server will listen to. For security purposes, configure firewall to allow access to port 5432 only for trusted IP addresses.

- `max_connections = 512`

  This is a minimum recommended value, you can specify a larger one.If you use a cluster Postgres DBMS, specify the `max_connections` parameter for all DBMS servers as well as in the cluster configuration.

- `shared_buffers = N`

  N = 25%–40% of the RAM value of the device where the DBMS is installed, but not less than 3 GB.

- `temp_buffers = 24MB`
- `work_mem = 16MB`
- `maintenance_work_mem = 128MB`
- `max_stack_depth = N`

  If the DBMS is installed on a Linux device, then N = maximum stack size minus the 1 MB safety margin. To obtain the maximum stack size value in KB, execute the '`ulimit -s`' command.

  If the DBMS is installed on a Windows device, then leave the default value 2 MB.

- `max_parallel_workers_per_gather = 0`
- `standard_conforming_strings` = on

If you use Postgres Pro 15.7 or Postgres Pro 15.7.1, disable the `enable_compound_index_stats` parameter:

```
enable_compound_index_stats = off
```

> Reload configuration or restart the DBMS server after updating the configuration file.

For detailed information about PostgreSQL and Postgres Pro server parameters and on how to specify the parameters, refer to the corresponding DBMS documentation.

Specify authentication entries for nodes in the pg_hba.conf configuration file. You can find it in the same path, where postgresql.conf is located.

- In the `"# "local" is for Unix domain socket connections only"` section, for `local`, in the `METHOD` column, change the value to `trust`
- In the `"IPv4 local connections"` section, for `host`, in the `ADDRESS` column, change the value to `0.0.0.0/0`
- Run the following command to apply the changes without restarting the database server:

  ```
  sudo -u postgres psql -c "SELECT pg_reload_conf();"
  ```

Specify the password for the `postgres` user:

1. Run the following commands:

   ```
   sudo su - postgres
   psql
   ```

   The command output should be as follows:

   ```
   psql (15.12)
   Type "help" for help.
   postgres=#
   ```

2. After `postgres=#`, enter `\password`:

   ```
   postgres=# \password
   ```

3. Enter and repeat the password:

   ```
   Enter new password for user "postgres"
   Enter it again:
   ```

4. To log out of a user `postgres`:

   ```
   postgres=# exit
   ```

Kaspersky Next XDR Expert components require PostgrSQL superuser credentials in `parameters.psql_dsn` as the `value`. Specify postgres credentials in the deployment configuration file. For more information, see Single-node deployment: Specifying the installation parameters or Multi-node deployment: Specifying the installation parameters sections depending on your deployment schema.

Checking the database connection

To check the database connection:

1. Make sure the Postgres service is running:

   ```
   systemctl list-unit-files | grep postgres
   ```

   Example output:

   ```
   postgrespro-std-17.service                                                  enabled         enabled
   ```

2. Restart the Postgres service:

   ```
   systemctl restart <Postgres_service>
   ```

   Where `<Postgres_service>` is the Postgres service name obtained at step 1 of this procedure.

3. Run `psql` to check the connection:

   ```
   psql -h <fqdn> -U postgres -d postgres
   ```

   Enter the password specified earlier:

   ```
   Password for user postgres:
   ```

   Example output:

   ```
   psql (17.4)
   Type "help" for help.
   ```

4. Enter `exit`:

   ```
   postgres=# exit
   ```

Highly available cluster support

Highly available PostgreSQL clusters are supported with the following settings:

- Replication type: synchronous
- Minimum number of nodes: 3
- Maximum number of nodes: 15

Set the `psql_dsn` parameter as follows: `postgres://<dbms_username>:<password>@<fqdn1>:<port>,<fqdn2>:<port>,<fqdn3>:<port>`

When using a Postgres Highly Available cluster version 14 or later as the database, the database user must have the `pg_read_all_stats` privilege. You can grant this privilege by running the following command:

```
GRANT pg_read_all_stats TO "dbms_username";
```

Where `dbms_username` is the name of a privileged internal DBMS account.

Specify the following parameter in the configuration file (`/etc/postgresql/<VERSION>/main/postgresql.conf`):

```
log_hostname = 1
```

Corosync/Pacemaker support

To use Corosync/Pacemaker, set the `psql_dsn` parameter with a virtual IP address of the DBMS cluster as follows:

```
psql_dsn=postgres://postgres:pass!@<virtual ip>:<port>
```

Preparing the KUMA inventory file for installing KUMA services outside the Kubernetes cluster

The KUMA inventory file is a file in the YAML format that contains installation parameters for deployment of the KUMA services that are not included in the Kubernetes cluster. The path to the KUMA inventory file is included in the configuration file that is used by Kaspersky Deployment Toolkit for the Kaspersky Next XDR Expert deployment.

The templates of the KUMA inventory file are located in the distribution package. If you want to install the KUMA services (storage, collector, and correlator) on one host, use the single.inventory.yml file. To install the services on several hosts in the network infrastructure, use the distributed.inventory.yml file.

We recommend backing up the KUMA inventory file that you used to install the KUMA services. You can use it to remove KUMA.

To prepare the KUMA inventory file,

Open the KUMA inventory file template located in the distribution package, and then edit the variables in the inventory file.

The KUMA inventory file contains the following blocks:

- `all` block

  The `all` block contains the variables that are applied to all hosts specified in the inventory file. The variables are located in the `vars` section.

- `kuma` block

  The `kuma` block contains the variables that are applied to hosts on which the KUMA services will be installed. These hosts are listed in the `kuma` block in the `children` section. The variables are located in the `vars` section.

The following table lists possible variables, their descriptions, possible values, and blocks of the KUMA inventory file where these variables can be located.

List of possible variables in the vars section

Variable	Description	Possible values	Block
Variables located in the `vars` section of the `all` and `kuma` blocks			
`ansible_connection`	Method used to connect to the KUMA service hosts.	• `ssh`—Connection to the target hosts via SSH is established. • `local`—No connection to the target hosts is established. To provide the correct installation of the KUMA services, in the `all` block, set the `ansible_connection` variable to `local`. In the `kuma` block, you must specify the `ansible_connection` variable and set `ansible_connection` to `ssh` to provide the connection to the hosts on which the KUMA services are installed via SSH.	• `all` • `kuma`
`ansible_user`	User name used to connect to KUMA service hosts to install external KUMA services.	If the root user is blocked on the target hosts, specify a user name that has the right to establish SSH connections and elevate privileges by using su or sudo. To provide the correct installation of the KUMA services, in the `all` block, set the `ansible_user` variable to `nonroot`. In the `kuma` block, you must override the `ansible_user` variable and set `ansible_user` to the username of the account that can connect to remote hosts via SSH, to prepare them for the installation of the KUMA services.	• `all` • `kuma`
`ansible_become`	Variable used to indicate the need to increase the privileges of the user account that is used to install KUMA components.	• `false`—If the `ansible_user` value is `root`. • `true`—If the `ansible_user` value is not `root`.	`kuma`
`ansible_become_method`	Method used for increasing the privileges of the user account that is used to install KUMA components.	`su` or `sudo` if the `ansible_user` value is not `root`.	`kuma`
Variables located in the `children` section of the `kuma` block			
`kuma_utils`	Group of hosts used for storing the service files and utilities of KUMA. A host can be included in the `kuma_utils` group and in the `kuma_collector`, `kuma_correlator`, or `kuma_storage` group at the same time. The `kuma_utils` group can contain multiple hosts. During the Kaspersky Next XDR Expert deployment, on the hosts that are included in `kuma_utils`, the following files are copied to the `/opt/kaspersky/kuma/utils/` directory: • kuma is an executable file with which the KUMA services are installed. • kuma.exe is an executable file with which the KUMA agents are installed on Windows-based hosts. • LEGAL_NOTICES is a file with information about third-party code. • maxpatrol-tool, kuma-ptvm.tar.gz are utilities for integration with MaxPatrol. • ootb-content is an archive with out of the box resources for the KUMA services.	The group of hosts contains the `ansible_host` variable that specifies the unique host FQDN and IP address.	`kuma`
`kuma_collector`	Group of KUMA collector hosts. This group can contain multiple hosts.	The group of KUMA collector hosts contains the `ansible_host` variable that specifies the unique host FQDN and IP address.	`kuma`

`kuma_correlator`	Group of KUMA correlator hosts. This group can contain multiple hosts.	The group of KUMA correlator hosts contains the `ansible_host` variable that specifies the unique host FQDN and IP address.	`kuma`
`kuma_storage`	Group of KUMA storage hosts. This group can contain multiple hosts.	The group of KUMA storage hosts contains the `ansible_host` variable that specifies the unique host FQDN and IP address.	`kuma`

Sample of the KUMA inventory file template for installation of the KUMA services on a single host (the single.inventory.yml file)

```
all:
  vars:
    ansible_connection: local
    ansible_user: nonroot
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_utils:
      hosts:
        kuma.example.com:
            ansible_host: 0.0.0.0
    kuma_collector:
      hosts:
        kuma.example.com:
            ansible_host: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma.example.com:
            ansible_host: 0.0.0.0
    kuma_storage:
      hosts:
        kuma.example.com:
            ansible_host: 0.0.0.0
            shard: 1
            replica: 1
            keeper: 1
```

Sample of the KUMA inventory file template for installation of the KUMA services on several hosts (the distributed.inventory.yml file)

```
all:
  vars:
    ansible_connection: local
    ansible_user: nonroot
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_utils:
      hosts:
        kuma-utils.example.com:
        ansible_host: 0.0.0.0
    kuma_collector:
```

```
      hosts:
        kuma-collector-1.example.com:
        ansible_host: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma-correlator-1.example.com:
        ansible_host: 0.0.0.0
    kuma_storage:
      hosts:
        kuma-storage-1.example.com:
            ansible_host: 0.0.0.0
            shard: 1
            replica: 1
            keeper: 1
        kuma-storage-2.example.com:
            ansible_host: 0.0.0.0
            shard: 1
            replica: 2
            keeper: 2
        kuma-storage-3.example.com:
            ansible_host: 0.0.0.0
            shard: 2
            replica: 1
            keeper: 3
        kuma-storage-4.example.com:
            ansible_host: 0.0.0.0
            shard: 2
            replica: 2
```

Multi-node deployment: Specifying the installation parameters

The configuration file is a file in the YAML format and contains a set of installation parameters for the Kaspersky Next XDR Expert components.

> The installation parameters listed in the tables below are required for the multi-node deployment of Kaspersky Next XDR Expert. To deploy Kaspersky Next XDR Expert on a single node, use the configuration file that contains the installation parameters specific for the single-node deployment.

The template of the configuration file (multinode.smp_param.yaml.template) is located in the distribution package in the archive with the KDT utility. You can fill out the configuration file template manually; or use the Configuration wizard to specify the installation parameters that are required for the Kaspersky Next XDR Expert deployment, and then generate the configuration file.

Not all of the parameters listed below are included in the configuration file template. This template contains only those parameters that must be specified before Kaspersky Next XDR Expert deployment. Remaining parameters are set to default values, and they are not included in the template. You can manually add these parameters to the configuration file to override its values. Parameter values must follow the rules described below. During the Kaspersky Next XDR Expert deployment, KDT checks whether the required installation parameters are specified correctly. If the required parameters in the configuration file are missing or invalid, the deployment will be interrupted.

Sample of the configuration file for the multi-node deployment of Kaspersky Next XDR Expert

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
# Nodes on which the Kubernetes cluster and Kaspersky XDR will be deployed
nodes:
  # Primary node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-primary1
    # Node type.
    # Possible node types: primary, worker.
    # Primary node is intended for managing the cluster, storing metadata, and
    distributing the workload.
    type: primary
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.42
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
  # Worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-w1
    # Node type.
    # Possible node types: primary, worker.
    # The workload of Kaspersky XDR components is distributed between worker nodes.
    type: worker
    # The KUMA service will be installed on the worker node. One, multiple, or no KUMA
    services can be specified.
    kuma_roles: ["core","metrics","collector"]
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.141
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
```

```
      # Password for connecting to the node. Can be used instead of the SSH key
      specified by the ssh_pk parameter.
      # password: pass
  # Worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-w2
    # Node type.
    # Possible node types: primary, worker.
    # The workload of Kaspersky XDR components is distributed between worker nodes.
    type: worker
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.45
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
  # Worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-w3
    # Node type.
    # Possible node types: primary, worker.
    # The workload of Kaspersky XDR components is distributed between worker nodes.
    type: worker
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.30
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
# Parameters for the Kaspersky XDR deployment
parameters:
  # Connection string for accessing the DBMS installed and configured on a separate
  server.
```

```
# user_db—The user name of a privileged internal DBMS account. This account is
granted permissions to create databases and other DBMS accounts.
# password_db—The password of the privileged internal DBMS account.
# fqdn_db:port_db—The FQDN and connection port of a separate server on which the
DBMS is installed.
# The 'psql_dsn' parameter value must comply with the URI format. If the connection
URI includes symbols with special meaning in any of its parts, it must be encoded
with percent-encoding.
- name: psql_dsn
  source:
    value: "postgres://postgres:password@dbms.example.com:1234"
# IP address of the Kubernetes cluster gateway.
# The Kubernetes cluster gateway is intended for connecting to the Kaspersky XDR
components installed inside the Kubernetes cluster.
# The gateway must be included in the same subnet as all cluster nodes.
- name: ingress_ip
  source:
    value: 10.80.23.182
# Path to the SSH private key for connection to the primary and workers nodes by
KDT. The SSH private key is located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: ssh_pk
  source:
    path: /root/.ssh/id_rsa
# Password of the Kaspersky XDR user account that will be created during the
installation.
# Password must be 8 to 256 characters long.
# It can contain lowercase or uppercase letters ('a' to 'z'), digits ('0' to '9'),
and special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( )
;).
- name: admin_password
  source:
    value: "password"
# Amount of disk space for the operation of KUMA Core, in gigabytes.
# Specified as "<amount>Gi". Default value is 512Gi.
# If the 'low_resources' parameter is set to true, the 'core_disk_request' parameter
is ignored and 4Gi is allocated.
- name: core_disk_request
  source:
    value: 512Gi
# Path to the KUMA inventory file that contains the installation parameters for
deployment of the KUMA services. This file is located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: inventory
  source:
    value: "/home/user/inventory.yml"
# Path to the Kaspersky Endpoint Detection and Response license key. This file is
located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: license
  source:
    value: "/home/user/license.key"
# Domain name. The parameter value must meet the requirements for second-level
domain naming.
# The service host name and domain name are parts of FQDNs of the Kaspersky XDR
services.
```

```
  # Default values of service host names: nwc_host—"console", flow_host—"console",
  hydra_host—"console", login_host—"console", ksc_host—"admsrv", console_host
  —"console", api_host—"api", kuma_host—"kuma", monitoring_host—"monitoring".
  # Custom host names can be specified in the configuration file.
  # Example: console.smp.local is the FQDN of the service that provides access to the
  OSMP Console. This FQDN consists of the 'console_host' and 'smp_domain' parameter
  values.
  - name: smp_domain
    source:
      value: "smp.local"
  # List of host names of the Kaspersky XDR services for which a self-signed or custom
  certificate must be generated.
  # The parameter value must be a list of host names, separated by spaces. If custom
  host names are not specified, the default host names are used.
  - name: pki_host_list
    source:
      value: "admsrv api console kuma monitoring"
```

For correct function of KDT with the configuration file, add an empty line at the end of the file.

The `nodes` section of the configuration file contains installation parameters for each target host of the Kubernetes cluster. These parameters are listed in the table below.

Nodes section

Parameter name	Required	Description

`desc`	Yes	The node name.
		The value must comply with the following rules:
		• The node name must be 1 to 63 characters long.
		• The node name can contain the following characters:
		  • Uppercase ASCII letters (A–Z)
		  • Lowercase ASCII letters (a–z)
		  • Numbers (0–9)
		  • Hyphen (-)
		All cluster nodes must have unique names. This requirement is one of the prerequisites for the installation of Kaspersky Next XDR Expert and is verified by KDT.

`type`	Yes	The node type. Possible parameter values: • `primary` • `worker` • `loadbalancer`
`host`	Yes	The IP address of the node. All nodes must be included in the same subnet as the Kubernetes cluster gateway. Nodes require IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on nodes and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks. If you have multiple primary nodes, a load balancer is required to distribute incoming requests across the primary nodes and ensure an even workload. If you want to use an external load balancer, allocate a separate host outside the cluster and specify its parameters in the configuration file. If you do not want to use an external load balancer, worker nodes will act as the load balancer.

Parameter name	Required	Description
`kuma_role`	No	The KUMA service that you can specify for installation on the worker node. You can specify one, multiple, or no KUMA services. The `kuma_roles` parameter is considered when the KUMA service installing in the Kubernetes cluster by using the OSMP Console. If the node where you plan to install the KUMA service is overloaded, the cluster will select another, less busy node for the service installation. The KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on worker nodes, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster. If all KUMA services will be installed inside the cluster, set the `inventory` parameter to `/dev/null`. It is not allowed to place storages created from the same KUMA resource on the same worker node. This rule ensures the required level of fault tolerance. If the storage cluster consists of multiple storage instances, all of them must be deployed either inside the cluster or outside the cluster. Possible parameter values: • `core` — KUMA Core • `metrics` — KUMA metrics • `collector` — Collector • `eventrouter` — Event router • `correlator` — Correlator • `storage` — Storage The parameter value is specified as a list of strings containing KUMA roles. For example: `kuma_roles: ["core","metrics","collector"]`

`user`	Yes	The user name of the account created on the target host and used for connection to the node by KDT.
		The value must comply with the following rules:
		• The user name must be 1 to 31 characters long.
		• The user name can contain the following characters:
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Underscore (_), hyphen (-)

`password`	No	The password for connecting to the node. Can be used instead of the SSH key specified by the `ssh_pk` parameter.

Other installation parameters are listed in the `parameters` section of the configuration file and are described in the table below.

Parameter name	Required	Description
psql_dsn	Yes	The connection string for accessing the DBMS that is installed and configured outside the Kubernetes cluster. Specify this parameter as follows: psql_dsn=postgres://<dbms_username>:<password>@<fqdn>:<port> where: • dbms_username —The user name of a privileged internal DBMS account. This account is granted permissions to create databases and other DBMS accounts. By using this privileged DBMS account, the databases and other DBMS accounts required for the Kaspersky Next XDR Expert components will be created during the deployment. • password —The password of the privileged internal DBMS account. • fqdn:port —The FQDN and connection port of the target host on which the DBMS is installed. To use a highly available cluster, specify this parameter as follows: psql_dsn=postgres://<dbms_username>:<password>@<fqdn1>:<port>,<fqdn2>:<port>,<fqdn3>:<port> To use Corosync/Pacemaker, specify this parameter with a virtual IP as follows: psql_dsn=postgres://postgres:pass!@<virtual ip>:<port> The psql_dsn parameter value must comply with the URI format. If the connection URI includes symbols with special meaning in any of its parts, it must be encoded with percent-encoding. Symbols that must be replaced in the psql_dsn parameter value: • Whitespace → %20 • ! → %21 • % → %25 • & → %26 • / → %2F • : → %3A • = → %3D • ? → %3F • @ → %40 • [→ %5B •] → %5D Refer to the PostgreSQL connection string article for details.
nwc-language	Yes	The language of the OSMP Console interface is specified by default. After installation, you can change the OSMP Console language. Possible parameter values: • enUS (default value) • ruRu The default value is used if you do not specify this parameter in the configuration file.
ingress_ip	Yes	The reserved static IPv4 address of the Kubernetes cluster gateway. The Kubernetes cluster gateway is intended for connecting to the Kaspersky Next XDR Expert components installed inside the Kubernetes cluster. The gateway must be included in the same subnet as the primary and worker nodes.
ssh_pk	Yes	The path to the private part of the SSH key located on the administrator host and used for connection to the primary worker node and nodes with the KUMA services (collectors, correlators, and storages) by using KDT. The parameter value must be a Linux file path or Base64-encoded file content.

Parameter name	Required	Description
`admin_password`	Yes	The `admin_password` parameter specifies the password of the Kaspersky Next XDR Expert user account that will be created by KDT during the installation. The default user name of this account is "admin". The Main administrator role is assigned to this user account. The password must comply with the following rules: • The user password cannot have fewer than 8 or more than 256 characters. • The password must contain characters from at least three of the groups listed below: • Uppercase letters (A–Z) • Lowercase letters (a–z) • Numbers (0–9) • Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;) • The password must not contain any whitespaces, or the ".@" combination. When you specify the `admin_password` parameter value manually (not by the Configuration wizard), make sure that this value meets the YAML standard requirements for values in strings: • The parameter value containing special characters must be enclosed in single quotes. • Any single quote ' inside the parameter value must be doubled to escape this single quote. Example: the user account password `Any_pass%1234'5678"90` must be specified as the value `'Any_pass%1234''5678"90'` of the `admin_password` parameter.
`low_resources`	No	The parameter indicating that Kaspersky Next XDR Expert is installed on the target host with limited computing resources.Set the `low_resources` parameter to `false` for multi-node deployment. Possible parameter values: • `true` —Installation with limited computing resources (for single-node deployment) • `false` —Standard installation (default value)
`core_disk_request`	Yes	The parameter that specifies the amount of disk space for the operation of KUMA Core. If you do not specify the `core_disk_request` parameter and the `low_resources` parameter is set to `false` or is not specified, the default amount of disk space for the operation of KUMA Core is 100 GB. If you do not specify the `core_disk_request` parameter and the `low_resources` parameter is set to `true`, the default amount of disk space for the operation of KUMA Core is 4 GB. You can change the `core_disk_request` parameter later, after the application deployment.
`metrics_disk_request`	Yes	The parameter that specifies the amount of disk space for the operation of the Metrics service. If you do not specify the `metrics_disk_request` parameter and the `low_resources` parameter is set to `false` or is not specified, the default amount of disk space for the operation of the Metrics service is 80 GB. If you do not specify the `metrics_disk_request` parameter and the `low_resources` parameter is set to `true`, the default amount of disk space for the operation of the Metrics service is 4 GB. You can change the `metrics_disk_request` parameter later, after the application deployment.
`inventory`	Yes	The path to the KUMA inventory file located on the administrator host. The inventory file contains installation parameters for deployment of the KUMA services that are not included in the Kubernetes cluster. The parameter value must be a Linux file path or Base64-encoded file content. If all KUMA services will be installed inside the cluster, set the `inventory` parameter to `/dev/null`.
`host_inventory`	No	The path to the additional KUMA inventory file located on the administrator host. This file contains the installation parameters used to partially add or remove hosts with the KUMA services. The parameter value must be a Linux file path or Base64-encoded file content. If you perform an initial deployment of Kaspersky Next XDR Expert or run a custom action that requires configuration file, leave the default parameter value (`/dev/null`).

Parameter name	Required	Description
license	Yes	The path to the KUMA license key located on the administrator host. The parameter value must be a Linux file path or Base64-encoded file content. You can upload the license key or specify the activation code once you log in to OSMP after deployment is complete.
nwc_host flow_host hydra_host login_host ksc_host console_host api_host kuma_host monitoring_host xdr_edr_ingress_host	Yes	The host name that is used in the FQDNs of the public Kaspersky Next XDR Expert services. The service host name and domain name (the smp_domain parameter value) are parts of the service FQDN. If custom host names are not specified, the following default host names are used: • nwc_host—"console" • flow_host—"console" • hydra_host—"console" • login_host—"console" • ksc_host—"admsrv" • console_host—"console" • api_host—"api" • kuma_host—"kuma" • monitoring_host—"monitoring" • xdr_edr_ingress_host—"agentserver"
smp_domain	Yes	The domain name that is used in the FQDNs of the public Kaspersky Next XDR Expert services. The parameter value must meet the requirements for second-level domain naming. The service host name and domain name are parts of the service FQDN. For example, if the value of the console_host variable is console, and the value of the smp_domain variable is smp.local, then the full name of the service that provides access to the OSMP Console is console.smp.local.
pki_host_list	Yes	The list of host names of the public Kaspersky Next XDR Expert services for which a self-signed or custom certificate is to be generated. The parameter value must be a list of host names, separated by spaces.
intermediate_enabled	No	The parameter that indicates whether to use the custom intermediate certificate instead of the self-signed certificates for the public Kaspersky Next XDR Expert services. Possible parameter values: • true—Use custom intermediate certificate (default value). • false—Use self-signed certificates.
intermediate_bundle	No	The path to the custom intermediate certificate used to work with public Kaspersky Next XDR Expert services. Specify this parameter if the intermediate_enabled parameter is set to true. The parameter value must be a Linux file path or Base64-encoded file content. The validity period for the custom intermediate certificate must be 399 days or more.
admsrv_bundle api_bundle console_bundle kuma_bundle monitoring_bundle	No	The paths to the custom leaf certificates used to work with the corresponding public Kaspersky Next XDR Expert services: <admsrv_host>.<smp_domain>, <api_host>.<smp_domain>, <kuma_host>.<smp_domain>, <monitoring_host>.<smp_domain>, and <console_host>.<smp_domain>. The parameter values must be a Linux file path or Base64-encoded file content. If you want to specify the custom leaf certificates, set the intermediate_enabled parameter to false and do not specify the intermediate_bundle parameter and the pki_host_list parameter.
encrypt_secret sign_secret	Yes	The names of the secret files that are stored in the Kubernetes cluster. These names contain the domain name, which must match the smp_domain parameter value.
ksc_state_size	Yes	The amount of free disk space allocated to store the Administration Server data (updates, installation packages, and other internal service data). Measured in gigabytes, specified as "<amount>Gi". The required amount of free disk space depends on the number of managed devices and other parameters, and can be calculated. The default value is 30 GB. This value is used if you do not specify this parameter in the configuration file.

Parameter name	Required	Description
`prometheus_size`	Yes	The amount of free disk space allocated to store metrics. Measured in gigabytes, specified as "<amount>GB". The minimum recommended and default value is 5 GB. The default value is used if you do not specify this parameter in the configuration file.

`grafana_admin_user`	No	The user name of the account used to view OSMP metrics through the Grafana tool.
		The value must comply with the following rules:
		• The user name must be 1 to 30 characters long.
		• The user name can contain the following characters:
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Special characters (_ - .)
		• The user name must start with a letter.
		• The user name must not contain either whitespaces or the "@" symbol.

`grafana_admin_password`	No	The password of the account used to view OSMP metrics through the Grafana tool.
		The password must comply with the following rules:
		• The user password cannot have fewer than 8 or more than 256 characters.
		• The password must contain characters from at least three of the groups listed below:
		  • Uppercase letters (A–Z)
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)
		• The password must not contain either whitespaces or ".@".

`loki_size`	Yes	The amount of free disk space allocated to store OSMP logs. Measured in gigabytes, specified as "<amount>Gi". The minimum recommended and default value is 20 GB. The default value is used if you do not specify this parameter in the configuration file.
`loki_retention_period`	Yes	The storage period of OSMP logs after which logs are automatically removed. The default value is 72 hours (set the parameter value in the configuration file as "<time in hours>h". For example, "72h"). The default value is used if you do not specify this parameter in the configuration file.
`file_storage_cp`	No	The amount of free disk space allocated to store data of the component for working with response actions. Measured in gigabytes, specified as "<amount>Gi". The minimum recommended value is 20 GB.
`psql_tls_off`	No	The parameter that indicates whether to encrypt the traffic between the Kaspersky Next XDR Expert components and the DBMS by using the TLS protocol. If the DBMS is installed outside the cluster, TLS encryption is disabled by default (the default value is `true`). Possible parameter values: • `true`—Do not encrypt the traffic (default value). • `false`—Encrypt the traffic.
`psql_trusted_cas`	No	The path to the PEM file that can contain the TLS certificate of the DBMS server or a root certificate from which the TLS server certificate can be issued. The parameter value must be a Linux file path or Base64-encoded file content. Specify the `psql_trusted_cas` parameter if the DBMS will be installed and configured on a separate server and traffic encryption is enabled (`psql_tls_off` is set to `false`).

Parameter name	Required	Description
`proxy_enabled`	No	The parameter that indicates whether to use the proxy server to connect the Kaspersky Next XDR Expert components to the internet. If the host on which Kaspersky Next XDR Expert is installed has internet access, you can also provide internet access for the operation of Kaspersky Next XDR Expert components (for example, Administration Server) and for specific integrations, both Kaspersky and third-party. To establish the proxy connection, you must also specify the proxy server parameters in the Administration Server properties. Possible parameter values: • `true` —Proxy server is used. • `false` —Proxy server is not used (default value).
`proxy_addresses`	No	The IP address of the proxy server. If the proxy server uses multiple IP addresses, specify these addresses separated by a space (for example, "`0.0.0.0 0.0.0.1 0.0.0.2`"). Specify this parameter if the `proxy_enabled` parameter is set to `true`.
`proxy_port`	No	The number of the port through which the proxy connection will be established. Specify this parameter if the `proxy_enabled` parameter is set to `true`.
`trace_level`	No	The verbosity level of Administration Server trace files. Traces are written during the Administration Server operation. The default value is 0, no traces are written. Possible parameter values: 0–5. The higher the parameter value, the more detailed the trace details.
`ansible_extra_flags`	No	The verbosity level of logs of the KUMA Core and KUMA services deployment that is performed by KDT. Possible parameter values: • `-v` • `-vv` • `-vvv` • `-vvvv` As the number of "v" letters in the flag increases, logs become more detailed. If this parameter is not specified in the configuration file, the standard component installation logs are saved.
`incident_attachments_max_count_limit`	No	The number of files that you can attach to the incident. The default value is `100`.
`incident_attachments_max_size_limit`	No	The total size of files attached to the incident. Measured in bytes. Specified without units of measurement. The default value is `26214400`.
`ignore_precheck`	No	The parameter indicating whether to check the hardware, software, and network configuration of the Kubernetes cluster nodes for compliance with the prerequisites for installing the solution before the deployment. Possible parameter values: • `true` —Skip the pre-checks. • `false` —Perform the pre-checks (default value).
`grafana_smtp_host`	No	SMTP server address for Grafana, used to send email notifications about Kubernetes cluster health events. Specify this parameter in the `<host>:<port>` format, for example, `localhost:25`.
`grafana_smtp_user`	No	User name for authentication on the SMTP server.
`grafana_smtp_password`	No	Password for authentication on the SMTP server.
`grafana_smtp_bundle`	No	File containing the X.509 certificate chain used for verifying TLS connection to the SMTP server.
`grafana_smtp_skip_verify`	No	The parameter that indicates whether to skip TLS certificate verification of the SMTP server. Possible parameter values: • `true` —Skip TLS certificate verification. • `false` —Perform TLS certificate verification (default value).
`grafana_smtp_from_address`	No	Email address of the sender, which will appear in the From field when sending email notifications.
`grafana_smtp_from_name`	No	Display name of the sender in email notifications.
`grafana_smtp_ehlo_identity`	No	Identifier used in the EHLO command when connecting to the SMTP server.

Parameter name	Required	Description
`grafana_starttls_policy`	No	Policy for using STARTTLS when connecting to the SMTP server. Possible parameter values: • `OpportunisticStartTLS` —Use STARTTLS if the SMTP server supports it; otherwise, connect to the SMPT server without encryption. • `MandatoryStartTLS` —Require STARTTLS; fail if the SMTP server does not support it. • `NoStartTLS` —Do not use STARTTLS; connect to the SMPT server without encryption. • `empty` —Use STARTTLS if the SMTP server supports it; otherwise, connect to the SMPT server without encryption (default value).
`grafana_smtp_enable_tracing`	No	The parameter that indicates whether to enable SMTP connection tracing for debugging email sending. Possible parameter values: • `true` —Enable SMTP connection tracing. • `false` —Disable SMTP connection tracing (default value).

Sample of the configuration file for the multi-node deployment of Kaspersky Next XDR Expert

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
# Nodes on which the Kubernetes cluster and Kaspersky XDR will be deployed
nodes:
  # Primary node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-primary1
    # Node type.
    # Possible node types: primary, worker.
    # Primary node is intended for managing the cluster, storing metadata, and
    distributing the workload.
    type: primary
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.42
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
  # Worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-w1
    # Node type.
    # Possible node types: primary, worker.
```

```
    # The workload of Kaspersky XDR components is distributed between worker nodes.
    type: worker
    # The KUMA service will be installed on the worker node. One, multiple, or no KUMA
    services can be specified.
    kuma_roles: ["core","metrics","collector"]
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.141
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
  # Worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-w2
    # Node type.
    # Possible node types: primary, worker.
    # The workload of Kaspersky XDR components is distributed between worker nodes.
    type: worker
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.45
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
  # Worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-w3
    # Node type.
    # Possible node types: primary, worker.
    # The workload of Kaspersky XDR components is distributed between worker nodes.
    type: worker
    # IP address of the node.
    # Primary and worker nodes must be included in the same subnet as the Kubernetes
    cluster gateway.
```

```
    host: 10.80.23.30
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
# Parameters for the Kaspersky XDR deployment
parameters:
  # Connection string for accessing the DBMS installed and configured on a separate
  server.
  # user_db—The user name of a privileged internal DBMS account. This account is
  granted permissions to create databases and other DBMS accounts.
  # password_db—The password of the privileged internal DBMS account.
  # fqdn_db:port_db—The FQDN and connection port of a separate server on which the
  DBMS is installed.
  # The 'psql_dsn' parameter value must comply with the URI format. If the connection
  URI includes symbols with special meaning in any of its parts, it must be encoded
  with percent-encoding.
  - name: psql_dsn
    source:
      value: "postgres://postgres:password@dbms.example.com:1234"
  # IP address of the Kubernetes cluster gateway.
  # The Kubernetes cluster gateway is intended for connecting to the Kaspersky XDR
  components installed inside the Kubernetes cluster.
  # The gateway must be included in the same subnet as all cluster nodes.
  - name: ingress_ip
    source:
      value: 10.80.23.182
  # Path to the SSH private key for connection to the primary and workers nodes by
  KDT. The SSH private key is located on the administrator host.
  # The parameter value must be a Linux file path or Base64-encoded file content.
  - name: ssh_pk
    source:
      path: /root/.ssh/id_rsa
  # Password of the Kaspersky XDR user account that will be created during the
  installation.
  # Password must be 8 to 256 characters long.
  # It can contain lowercase or uppercase letters ('a' to 'z'), digits ('0' to '9'),
  and special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( )
  ;).
  - name: admin_password
    source:
      value: "password"
  # Amount of disk space for the operation of KUMA Core, in gigabytes.
  # Specified as "<amount>Gi". Default value is 512Gi.
  # If the 'low_resources' parameter is set to true, the 'core_disk_request' parameter
  is ignored and 4Gi is allocated.
  - name: core_disk_request
    source:
      value: 512Gi
```

```
  # Path to the KUMA inventory file that contains the installation parameters for
  deployment of the KUMA services. This file is located on the administrator host.
  # The parameter value must be a Linux file path or Base64-encoded file content.
  - name: inventory
    source:
      value: "/home/user/inventory.yml"
  # Path to the Kaspersky Endpoint Detection and Response license key. This file is
  located on the administrator host.
  # The parameter value must be a Linux file path or Base64-encoded file content.
  - name: license
    source:
      value: "/home/user/license.key"
  # Domain name. The parameter value must meet the requirements for second-level
  domain naming.
  # The service host name and domain name are parts of FQDNs of the Kaspersky XDR
  services.
  # Default values of service host names: nwc_host—"console", flow_host—"console",
  hydra_host—"console", login_host—"console", ksc_host—"admsrv", console_host
  —"console", api_host—"api", kuma_host—"kuma", monitoring_host—"monitoring".
  # Custom host names can be specified in the configuration file.
  # Example: console.smp.local is the FQDN of the service that provides access to the
  OSMP Console. This FQDN consists of the 'console_host' and 'smp_domain' parameter
  values.
  - name: smp_domain
    source:
      value: "smp.local"
  # List of host names of the Kaspersky XDR services for which a self-signed or custom
  certificate must be generated.
  # The parameter value must be a list of host names, separated by spaces. If custom
  host names are not specified, the default host names are used.
  - name: pki_host_list
    source:
      value: "admsrv api console kuma monitoring"
```

Single-node deployment: Specifying the installation parameters

The configuration file used to deploy Kaspersky Next XDR Expert on a single node contains the installation parameters that are required both for the multi-node and single-node deployment. Also, this configuration file contains parameters specific only for the single-node deployment (`openbao_ha_mode`, `openbao_standalone`, and `default_class_replica_count`).

The template of the configuration file (singlenode.smp_param.yaml.template) is located in the distribution package in the archive with the KDT utility. You can fill out the configuration file template manually; or use the Configuration wizard to specify the installation parameters that are required for the Kaspersky Next XDR Expert deployment, and then generate the configuration file.

Not all of the parameters listed below are included in the configuration file template. This template contains only those parameters that must be specified before Kaspersky Next XDR Expert deployment. The remaining parameters are set to default values, and they are not included in the template. You can manually add these parameters to the configuration file to override its values. Parameter values must follow the rules described below. During the Kaspersky Next XDR Expert deployment, KDT checks whether the required installation parameters are specified correctly. If the required parameters in the configuration file are missing or invalid, the deployment will be interrupted.

Sample of the configuration file for the single-node deployment of Kaspersky Next XDR Expert

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
# Node on which the Kubernetes cluster and Kaspersky XDR will be deployed
nodes:
  # Primary worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-1
    # Node type.
    # For the single-node deployment, set the type parameter to primary-worker. In
    this case, the target host will act as the primary and worker nodes.
    type: primary-worker
    # IP address of the node.
    # The primary worker node must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.42
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
# Parameters for the Kaspersky XDR deployment
parameters:
  # Connection string for accessing the DBMS installed and configured on a separate
  server.
  # user_db—The user name of a privileged internal DBMS account. This account is
  granted permissions to create databases and other DBMS accounts.
  # password_db—The password of the privileged internal DBMS account.
  # fqdn_db:port_db—The FQDN and connection port of a separate server on which the
  DBMS is installed.
  # The 'psql_dsn' parameter value must comply with the URI format. If the connection
  URI includes symbols with special meaning in any of its parts, it must be encoded
  with percent-encoding.
  - name: psql_dsn
    source:
      value: "postgres://postgres:password@dbms.example.com:1234"
  # IP address of the Kubernetes cluster gateway.
  # The Kubernetes cluster gateway is intended for connecting to the Kaspersky XDR
  components installed inside the Kubernetes cluster.
  # The gateway must be included in the same subnet as the primary worker node.
  - name: ingress_ip
    source:
      value: 10.80.23.182
  # Path to the SSH private key for connection to the primary worker node by KDT. The
  SSH private key is located on the administrator host.
  # The parameter value must be a Linux file path or Base64-encoded file content.
  - name: ssh_pk
```

```
  source:
    path: /root/.ssh/id_rsa
# Password of the Kaspersky XDR user account that will be created during the
installation.
# Password must be 8 to 256 characters long.
# It can contain lowercase or uppercase letters ('a' to 'z'), digits ('0' to '9'),
and special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( )
;).
- name: admin_password
  source:
    value: ""
# The parameter indicating that Kaspersky XDR is installed on the target host with
limited computing resources.
# For the single-node deployment, set the 'low_resources' parameter to "true" so
that Kaspersky Next XDR Expert components will require less memory and CPU
resources.
# If you enable this parameter, 4 GB of free disk space will be allocated to install
KUMA Core on the target host.
- name: low_resources
  source:
    value: "true"
# The number of disk volumes that are used to store the service data of Kaspersky
XDR components and KDT.
# For the single-node deployment, set the parameter value to "1".
- name: default_class_replica_count
  source:
    value: "1"
# The parameter indicating whether to run the secret storage in the High
Availability mode.
# For the single-node deployment, set the parameter value to "false".
- name: openbao_ha_mode
  source:
    value: "false"
# The parameter indicating whether to run the secret storage in the standalone
mode.
# For the single-node deployment, set the parameter value to "true".
- name: openbao_standalone
  source:
    value: "true"
# Path to the KUMA inventory file that contains the installation parameters for
deployment of the KUMA services. This file is located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: inventory
  source:
    value: "/home/user/inventory.yml"
# Path to the Kaspersky Endpoint Detection and Response license key. This file is
located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: license
  source:
    value: "/home/user/license.key"
# Domain name. The parameter value must meet the requirements for second-level
domain naming.
# The service host name and domain name are parts of FQDNs of the Kaspersky XDR
services.
# Default values of service host names: nwc_host—"console", flow_host—"console",
hydra_host—"console", login_host—"console", ksc_host—"admsrv", console_host
—"console", api_host—"api", kuma_host—"kuma", monitoring_host—"monitoring".
```

```
  # Custom host names can be specified in the configuration file.
  # Example: console.smp.local is the FQDN of the service that provides access to the
  OSMP Console. This FQDN consists of the 'console_host' and 'smp_domain' parameter
  values.
  - name: smp_domain
    source:
      value: "smp.local"
  # List of host names of the Kaspersky XDR services for which a self-signed or custom
  certificate must be generated.
  # The parameter value must be a list of host names, separated by spaces. If custom
  host names are not specified, the default host names are used.
  - name: pki_host_list
    source:
      value: "admsrv api console kuma monitoring"
```

For correct function of KDT with the configuration file, add an empty line at the end of the file.

The `nodes` section of the configuration file contains the target host parameters that are listed in the table below.

Nodes section

Parameter name	Required	Description

`desc`	Yes	The node name.
		The value must comply with the following rules:
		• The node name must be 1 to 63 characters long.
		• The node name can contain the following characters:
		  • Uppercase ASCII letters (A–Z)
		  • Lowercase ASCII letters (a–z)
		  • Numbers (0–9)
		  • Hyphen (-)
		All cluster nodes must have unique names. This requirement is one of the prerequisites for the installation of Kaspersky Next XDR Expert and is verified by KDT.

`type`	Yes	The node type. For the target host, set the `type` parameter to `primary-worker` to enable the **single-node deployment**. In this case, the target host will act as the primary and worker nodes.
`host`	Yes	The IP address of the node. The primary worker node must be included in the same subnet as the Kubernetes cluster gateway. The node requires IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on the node and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks.
`kuma_role`	No	The KUMA service that you can specify for installation on the worker node. KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on worker nodes, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster. If all KUMA services will be installed inside the cluster, set the `inventory` parameter to `/dev/null`. For **single-node deployment**, leave this parameter empty, because all services will be installed on a single node.

Parameter name	Required	Description

`user`	Yes	The user name of the account created on the target host and used for connection to the node by KDT.
		The value must comply with the following rules:
		• The user name must be 1 to 31 characters long.
		• The user name can contain the following characters:
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Underscore (_), hyphen (-)

`password`	No	The password for connecting to the node. Can be used instead of the SSH key specified by the `ssh_pk` parameter.

Other installation parameters are listed in the `parameters` section of the configuration file and are described in the table below.

Parameters section

Parameter name	Required	Description

`psql_dsn`	Yes	The connection string for accessing the DBMS that is installed and configured outside the Kubernetes cluster.
		Specify this parameter as follows:
		`psql_dsn=postgres://<dbms_username>:<password>@<fqdn>:<port>`
		where:
		• `dbms_username` —The user name of a privileged internal DBMS account. This account is granted permissions to create databases and other DBMS accounts. By using this privileged DBMS account, the databases and other DBMS accounts required for the Kaspersky Next XDR Expert components will be created during the deployment.
		• `password` —The password of the privileged internal DBMS account.
		• `fqdn:port` —The FQDN and connection port of the target host on which the DBMS is installed.
		If your DBMS is installed on the same host as a Kubernetes cluster node, do not use `localhost` or `127.0.0.1` as the DBMS address in the `psql_dsn` parameter. Use the host's IP address that is reachable over the cluster network (not a loopback address).
		To use a highly available cluster, specify this parameter as follows:
		`psql_dsn=postgres://<dbms_username>:<password>@<fqdn1>:<port>,<fqdn2>:<port>,<fqdn3>:<port>`
		To use Corosync/Pacemaker, specify this parameter with a virtual IP as follows:
		`psql_dsn=postgres://postgres:pass!@<virtual ip>:<port>`
		The `psql_dsn` parameter value must comply with the URI format. If the connection URI includes symbols with special meaning in any of its parts, it must be encoded with percent-encoding.
		Symbols that must be replaced in the `psql_dsn` parameter value:
		• Whitespace → `%20`
		• ! → `%21`
		• % → `%25`
		• & → `%26`
		• / → `%2F`
		• : → `%3A`
		• = → `%3D`
		• ? → `%3F`
		• @ → `%40`
		• [→ `%5B`
		•] → `%5D`
		Refer to the PostgreSQL connection string article for details.

Parameter name	Required	Description
`nwc-language`	Yes	The language of the OSMP Console interface is specified by default. After installation, you can change the OSMP Console language. Possible parameter values: • `enUS` (default value) • `ruRu` The default value is used if you do not specify this parameter in the configuration file.
`ingress_ip`	Yes	The reserved static IPv4 address of the Kubernetes cluster gateway. The Kubernetes cluster gateway is intended for connecting to the Kaspersky Next XDR Expert components installed inside the Kubernetes cluster. The gateway must be included in the same subnet as the primary worker node. The Kubernetes cluster gateway requires IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on the node and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks. The value of this parameter cannot be the IP address of the same host on which the deployment is being performed.
`ssh_pk`	Yes	The path to the private part of the SSH key located on the administrator host and used for connection to the primary worker node and nodes with the KUMA services (collectors, correlators, and storages) by using KDT. The parameter value must be a Linux file path or Base64-encoded file content.
`admin_password`	Yes	The `admin_password` parameter specifies the password of the Kaspersky Next XDR Expert user account that will be created by KDT during the installation. The default user name of this account is "admin". The Main administrator role is assigned to this user account. The password must comply with the following rules: • The user password cannot have fewer than 8 or more than 256 characters. • The password must contain characters from at least three of the groups listed below: • Uppercase letters (A–Z) • Lowercase letters (a–z) • Numbers (0–9) • Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;) • The password must not contain any whitespaces, or the ".@" combination. When you specify the `admin_password` parameter value manually (not by the Configuration wizard), make sure that this value meets the YAML standard requirements for values in strings: • The parameter value containing special characters must be enclosed in single quotes. • Any single quote ' inside the parameter value must be doubled to escape this single quote. Example: the user account password `Any_pass%1234'5678"90` must be specified as the value `'Any_pass%1234''5678"90'` of the `admin_password` parameter.
`low_resources`	Yes	The parameter indicating that Kaspersky Next XDR Expert is installed on the target host with limited computing resources. Possible parameter values: • `true` —Installation with limited computing resources (for single-node deployment) • `false` —Standard installation For the **single-node deployment**, set the `low_resources` parameter to `true` so that Kaspersky Next XDR Expert components will require less memory and CPU resources. Also, if you enable this parameter, 4 GB of free disk space will be allocated to install KUMA Core on the target host.
`openbao_ha_mode`	Yes	The parameter that indicates whether to run the secret storage in the High Availability (HA) mode. Possible parameter values: • `true` • `false` For the **single-node deployment**, set the `openbao_ha_mode` parameter to `false`.

Parameter name	Required	Description
`openbao_standalone`	Yes	The parameter that indicates whether to run the secret storage in the standalone mode. Possible parameter values: • `true` • `false` For the **single-node deployment**, set the `openbao_standalone` parameter value to `true`.
`default_class_replica_count`	Yes	The number of disk volumes that are used to store the service data of Kaspersky Next XDR Expert components and KDT. The default value is `3`. For the **single-node deployment**, set the `default_class_replica_count` parameter value to `1`.
`core_disk_request`	Yes	The parameter that specifies the amount of disk space for the operation of KUMA Core. If you do not specify the `core_disk_request` parameter and the `low_resources` parameter is set to `false` or is not specified, the default amount of disk space for the operation of KUMA Core is 100 GB. If you do not specify the `core_disk_request` parameter and the `low_resources` parameter is set to `true`, the default amount of disk space for the operation of KUMA Core is 4 GB. You can change the `core_disk_request` parameter later, after the application deployment.
`metrics_disk_request`	Yes	The parameter that specifies the amount of disk space for the operation of the Metrics service. If you do not specify the `metrics_disk_request` parameter and the `low_resources` parameter is set to `false` or is not specified, the default amount of disk space for the operation of the Metrics service is 80 GB. If you do not specify the `metrics_disk_request` parameter and the `low_resources` parameter is set to `true`, the default amount of disk space for the operation of the Metrics service is 4 GB. You can change the `metrics_disk_request` parameter later, after the application deployment.
`inventory`	Yes	The path to the KUMA inventory file located on the administrator host. The inventory file contains installation parameters for deployment of the KUMA services that are not included in the Kubernetes cluster. The parameter value must be a Linux file path or Base64-encoded file content. If all KUMA services will be installed inside the cluster, set the `inventory` parameter to `/dev/null`.
`host_inventory`	No	The path to the additional KUMA inventory file located on the administrator host. This file contains the installation parameters used to partially add or remove hosts with the KUMA services. The parameter value must be a Linux file path or Base64-encoded file content. If you perform an initial deployment of Kaspersky Next XDR Expert or run a custom action that requires configuration file, leave the default parameter value (`/dev/null`).
`license`	Yes	The path to the KUMA license key located on the administrator host. The parameter value must be a Linux file path or Base64-encoded file content. You can upload the license key or specify the activation code once you log in to OSMP after deployment is complete.

Parameter name	Required	Description

`nwc_host`
`flow_host`
`hydra_host`
`login_host`
`ksc_host`
`console_host`
`api_host`
`kuma_host`
`monitoring_host`
`xdr_edr_ingress_hos`
	Yes	The host name that is used in the FQDNs of the public Kaspersky Next XDR Expert services. The service host name and domain name (the `smp_domain` parameter value) are parts of the service FQDN.
		If custom host names are not specified, the following default host names are used:
		• `nwc_host—"console"`
		• `flow_host—"console"`
		• `hydra_host—"console"`
		• `login_host—"console"`
		• `ksc_host—"admsrv"`
		• `console_host—"console"`
		• `api_host—"api"`
		• `kuma_host—"kuma"`
		• `monitoring_host—"monitoring"`
		• `xdr_edr_ingress_host—"agentserver"`

`smp_domain`	Yes	The domain name that is used in the FQDNs of the public Kaspersky Next XDR Expert services. The parameter value must meet the requirements for second-level domain naming. The service host name and domain name are parts of the service FQDN. For example, if the value of the `console_host` variable is `console`, and the value of the `smp_domain` variable is `smp.local`, then the full name of the service that provides access to the OSMP Console is `console.smp.local`.
`pki_host_list`	Yes	The list of host names of the public Kaspersky Next XDR Expert services for which a self-signed or custom certificate is to be generated. The parameter value must be a list of host names, separated by spaces.
`intermediate_enabled`	No	The parameter that indicates whether to use the custom intermediate certificate instead of the self-signed certificates for the public Kaspersky Next XDR Expert services. Possible parameter values: • `true`—Use custom intermediate certificate (default value). • `false`—Use self-signed certificates.
`intermediate_bundle`	No	The path to the custom intermediate certificate used to work with public Kaspersky Next XDR Expert services. Specify this parameter if the `intermediate_enabled` parameter is set to `true`. The parameter value must be a Linux file path or Base64-encoded file content. The validity period for the custom intermediate certificate must be 399 days or more.

`admsrv_bundle`
`api_bundle`
`console_bundle`
`kuma_bundle`
`monitoring_bundle`
	No	The paths to the custom leaf certificates used to work with the corresponding public Kaspersky Next XDR Expert services: `<admsrv_host>.<smp_domain>`, `<api_host>.<smp_domain>`, `<kuma_host>.<smp_domain>`, `<monitoring_host>.<smp_domain>`, and `<console_host>.<smp_domain>`. The parameter values must be a Linux file path or Base64-encoded file content.
		If you want to specify the custom leaf certificates, set the `intermediate_enabled` parameter to `false` and do not specify the `intermediate_bundle` parameter and the `pki_host_list` parameter.

`encrypt_secret` `sign_secret`	Yes	The names of the secret files that are stored in the Kubernetes cluster. These names contain the domain name, which must match the `smp_domain` parameter value.
`ksc_state_size`	Yes	The amount of free disk space allocated to store the Administration Server data (updates, installation packages, and other internal service data). Measured in gigabytes, specified as "<amount>Gi". The required amount of free disk space depends on the number of managed devices and other parameters, and can be calculated. The default value is 30 GB. This value is used if you do not specify this parameter in the configuration file.
`prometheus_size`	Yes	The amount of free disk space allocated to store metrics. Measured in gigabytes, specified as "<amount>GB". The minimum recommended and default value is 5 GB. The default value is used if you do not specify this parameter in the configuration file.

Parameter name	Required	Description

`grafana_admin_user`	No	The user name of the account used to view OSMP metrics through the Grafana tool.
		The value must comply with the following rules:
		• The user name must be 1 to 30 characters long.
		• The user name can contain the following characters:
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Special characters (_ - .)
		• The user name must start with a letter.
		• The user name must not contain either whitespaces or the "@" symbol.

`grafana_admin_password`	No	The password of the account used to view OSMP metrics through the Grafana tool.
		The password must comply with the following rules:
		• The user password cannot have fewer than 8 or more than 256 characters.
		• The password must contain characters from at least three of the groups listed below:
		  • Uppercase letters (A–Z)
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)
		• The password must not contain either whitespaces or ".@".

`loki_size`	Yes	The amount of free disk space allocated to store OSMP logs. Measured in gigabytes, specified as "<amount>Gi". The minimum recommended and default value is 20 GB. The default value is used if you do not specify this parameter in the configuration file.
`loki_retention_period`	Yes	The storage period of OSMP logs after which logs are automatically removed. The default value is 72 hours (set the parameter value in the configuration file as "<time in hours>h". For example, "72h"). The default value is used if you do not specify this parameter in the configuration file.
`file_storage_cp`	No	The amount of free disk space allocated to store data of the component for working with response actions. Measured in gigabytes, specified as "<amount>Gi". The minimum recommended value is 20 GB.
`psql_tls_off`	No	The parameter that indicates whether to encrypt the traffic between the Kaspersky Next XDR Expert components and the DBMS by using the TLS protocol. If the DBMS is installed outside the cluster, TLS encryption is disabled by default (the default value is `true`). Possible parameter values: • `true`—Do not encrypt the traffic (default value). • `false`—Encrypt the traffic.
`psql_trusted_cas`	No	The path to the PEM file that can contain the TLS certificate of the DBMS server or a root certificate from which the TLS server certificate can be issued. The parameter value must be a Linux file path or Base64-encoded file content. Specify the `psql_trusted_cas` parameter if the DBMS will be installed and configured on a separate server and traffic encryption is enabled (`psql_tls_off` is set to `false`).
`proxy_enabled`	No	The parameter that indicates whether to use the proxy server to connect the Kaspersky Next XDR Expert components to the internet. If the host on which Kaspersky Next XDR Expert is installed has internet access, you can also provide internet access for the operation of Kaspersky Next XDR Expert components (for example, Administration Server) and for specific integrations, both Kaspersky and third-party. To establish the proxy connection, you must also specify the proxy server parameters in the Administration Server properties. Possible parameter values: • `true`—Proxy server is used. • `false`—Proxy server is not used (default value).

Parameter name	Required	Description
`proxy_addresses`	No	The IP address of the proxy server. If the proxy server uses multiple IP addresses, specify these addresses separated by a space (for example, "`0.0.0.0 0.0.0.1 0.0.0.2`"). Specify this parameter if the `proxy_enabled` parameter is set to `true`.
`proxy_port`	No	The number of the port through which the proxy connection will be established. Specify this parameter if the `proxy_enabled` parameter is set to `true`.
`trace_level`	No	The verbosity level of Administration Server trace files. Traces are written during the Administration Server operation. The default value is 0, no traces are written. Possible parameter values: 0–5. The higher the parameter value, the more detailed the trace details.
`ansible_extra_flags`	No	The verbosity level of logs of the KUMA Core and KUMA services deployment that is performed by KDT. Possible parameter values: • `-v` • `-vv` • `-vvv` • `-vvvv` As the number of "v" letters in the flag increases, logs become more detailed. If this parameter is not specified in the configuration file, the standard component installation logs are saved.
`incident_attachments_max_count_limit`	No	The number of files that you can attach to the incident. The default value is `100`.
`incident_attachments_max_size_limit`	No	The total size of files attached to the incident. Measured in bytes. Specified without units of measurement. The default value is `26214400`.
`ignore_precheck`	No	The parameter indicating whether to check the hardware, software, and network configuration of the Kubernetes cluster nodes for compliance with the prerequisites for installing the solution before the deployment. Possible parameter values: • `true` —Skip the pre-checks. • `false` —Perform the pre-checks (default value).
`grafana_smtp_host`	No	SMTP server address for Grafana, used to send email notifications about Kubernetes cluster health events. Specify this parameter in the `<host>:<port>` format, for example, `localhost:25`.
`grafana_smtp_user`	No	User name for authentication on the SMTP server.
`grafana_smtp_password`	No	Password for authentication on the SMTP server.
`grafana_smtp_bundle`	No	File containing the X.509 certificate chain used for verifying TLS connection to the SMTP server.
`grafana_smtp_skip_verify`	No	The parameter that indicates whether to skip TLS certificate verification of the SMTP server. Possible parameter values: • `true` —Skip TLS certificate verification. • `false` —Perform TLS certificate verification (default value).
`grafana_smtp_from_address`	No	Email address of the sender, which will appear in the From field when sending email notifications.
`grafana_smtp_from_name`	No	Display name of the sender in email notifications.
`grafana_smtp_ehlo_identity`	No	Identifier used in the EHLO command when connecting to the SMTP server.
`grafana_starttls_policy`	No	Policy for using STARTTLS when connecting to the SMTP server. Possible parameter values: • `OpportunisticStartTLS` —Use STARTTLS if the SMTP server supports it; otherwise, connect to the SMPT server without encryption. • `MandatoryStartTLS` —Require STARTTLS; fail if the SMTP server does not support it. • `NoStartTLS` —Do not use STARTTLS; connect to the SMPT server without encryption. • `empty` —Use STARTTLS if the SMTP server supports it; otherwise, connect to the SMPT server without encryption (default value).

Parameter name	Required	Description
`grafana_smtp_enable_tracing`	No	The parameter that indicates whether to enable SMTP connection tracing for debugging email sending. Possible parameter values: • `true` —Enable SMTP connection tracing. • `false` —Disable SMTP connection tracing (default value).

Sample of the configuration file for the single-node deployment of Kaspersky Next XDR Expert

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
# Node on which the Kubernetes cluster and Kaspersky XDR will be deployed
nodes:
  # Primary worker node parameters
    # Node name. The value must be 1 to 63 characters long.
    # It can only contain the ASCII letters 'a' to 'z' (in either lowercase or
    uppercase), the digits '0' to '9', and the hyphen ('-').
  - desc: cdt-1
    # Node type.
    # For the single-node deployment, set the type parameter to primary-worker. In
    this case, the target host will act as the primary and worker nodes.
    type: primary-worker
    # IP address of the node.
    # The primary worker node must be included in the same subnet as the Kubernetes
    cluster gateway.
    host: 10.80.23.42
    access:
      ssh:
        # User name of the account.
        # The user account is created on the target host and is used by KDT to connect
        to the node.
        # The user name must be 1 to 31 characters long.
        # It can contain letters ('a' to 'z'), digits ('0' to '9'), underscores ('_'),
        and hyphens ('-').
        user: root
        # Password for connecting to the node. Can be used instead of the SSH key
        specified by the ssh_pk parameter.
        # password: pass
# Parameters for the Kaspersky XDR deployment
parameters:
  # Connection string for accessing the DBMS installed and configured on a separate
  server.
  # user_db—The user name of a privileged internal DBMS account. This account is
  granted permissions to create databases and other DBMS accounts.
  # password_db—The password of the privileged internal DBMS account.
  # fqdn_db:port_db—The FQDN and connection port of a separate server on which the
  DBMS is installed.
  # The 'psql_dsn' parameter value must comply with the URI format. If the connection
  URI includes symbols with special meaning in any of its parts, it must be encoded
  with percent-encoding.
  - name: psql_dsn
    source:
      value: "postgres://postgres:password@dbms.example.com:1234"
  # IP address of the Kubernetes cluster gateway.
```

```
# The Kubernetes cluster gateway is intended for connecting to the Kaspersky XDR
components installed inside the Kubernetes cluster.
# The gateway must be included in the same subnet as the primary worker node.
- name: ingress_ip
  source:
    value: 10.80.23.182
# Path to the SSH private key for connection to the primary worker node by KDT. The
SSH private key is located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: ssh_pk
  source:
    path: /root/.ssh/id_rsa
# Password of the Kaspersky XDR user account that will be created during the
installation.
# Password must be 8 to 256 characters long.
# It can contain lowercase or uppercase letters ('a' to 'z'), digits ('0' to '9'),
and special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( )
;).
- name: admin_password
  source:
    value: ""
# The parameter indicating that Kaspersky XDR is installed on the target host with
limited computing resources.
# For the single-node deployment, set the 'low_resources' parameter to "true" so
that Kaspersky Next XDR Expert components will require less memory and CPU
resources.
# If you enable this parameter, 4 GB of free disk space will be allocated to install
KUMA Core on the target host.
- name: low_resources
  source:
    value: "true"
# The number of disk volumes that are used to store the service data of Kaspersky
XDR components and KDT.
# For the single-node deployment, set the parameter value to "1".
- name: default_class_replica_count
  source:
    value: "1"
# The parameter indicating whether to run the secret storage in the High
Availability mode.
# For the single-node deployment, set the parameter value to "false".
- name: openbao_ha_mode
  source:
    value: "false"
# The parameter indicating whether to run the secret storage in the standalone
mode.
# For the single-node deployment, set the parameter value to "true".
- name: openbao_standalone
  source:
    value: "true"
# Path to the KUMA inventory file that contains the installation parameters for
deployment of the KUMA services. This file is located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
- name: inventory
  source:
    value: "/home/user/inventory.yml"
# Path to the Kaspersky Endpoint Detection and Response license key. This file is
located on the administrator host.
# The parameter value must be a Linux file path or Base64-encoded file content.
```

```
  - name: license
    source:
      value: "/home/user/license.key"
  # Domain name. The parameter value must meet the requirements for second-level
  domain naming.
  # The service host name and domain name are parts of FQDNs of the Kaspersky XDR
  services.
  # Default values of service host names: nwc_host—"console", flow_host—"console",
  hydra_host—"console", login_host—"console", ksc_host—"admsrv", console_host
  —"console", api_host—"api", kuma_host—"kuma", monitoring_host—"monitoring".
  # Custom host names can be specified in the configuration file.
  # Example: console.smp.local is the FQDN of the service that provides access to the
  OSMP Console. This FQDN consists of the 'console_host' and 'smp_domain' parameter
  values.
  - name: smp_domain
    source:
      value: "smp.local"
  # List of host names of the Kaspersky XDR services for which a self-signed or custom
  certificate must be generated.
  # The parameter value must be a list of host names, separated by spaces. If custom
  host names are not specified, the default host names are used.
  - name: pki_host_list
    source:
      value: "admsrv api console kuma monitoring"
```

Specifying the installation parameters by using the Configuration wizard

For the multi-node and single-node Kaspersky Next XDR Expert deployment, you have to prepare a configuration file that contains the installation parameters of the Kaspersky Next XDR Expert components. The Configuration wizard allows you to specify the installation parameters that are required to deploy Kaspersky Next XDR Expert, and then generate the resulting configuration file. Optional installation parameters have default values, and they are not to be specified in the Configuration wizard. You can manually add these parameters to the configuration file to override their default values.

Prerequisites

Before specifying the installation parameters by using the Configuration wizard, you must install a database management system on a separate server that is located outside the Kubernetes cluster, perform all preparatory steps necessary for the administrator, target hosts (depending on the multi-node or single-node deployment option), and KUMA hosts.

Process

To specify the installation parameters by using the Configuration wizard:

1. On the administrator host where the KDT utility is located, run the Configuration wizard by using the following command:

   ```
   ./kdt wizard -k <path_to_transport_archive> -o <path_to_configuration_file>
   ```

where:

- < `path_to_transport_archive` > is the path to the transport archive.
- < `path_to_configuration_file` > is the path where you want to save the configuration file and the configuration file name.

The Configuration wizard prompts you to specify the installation parameters. The list of the installation parameters that are specific for the multi-node and single-node deployment differs.

If you do not have the Write permissions on the specified directory or a file with the same name is located in this directory, an error occurs and the wizard terminates.

2. Enter the number of primary nodes.

   Possible values:

   - 1—Single-node deployment or multi-node deployment.
   - 2 or more—Multi-node deployment.

   The Kubernetes cluster can have multiple primary nodes only in the case of a multi-node deployment.

3. Enter the IPv4 address of a primary node (or a primary worker node, if you will perform the single-node deployment). This value corresponds to the `host` parameter of the configuration file.

   This node must be included in the same subnet as the Kubernetes cluster gateway.

4. Enter the user name of the account used for connection to the primary node by KDT (the `user` parameter of the configuration file).

   The value must comply with the following rules:

   - The user name must be 1 to 31 characters long.
   - The user name can contain the following characters:
     - Lowercase letters (a–z)
     - Numbers (0–9)
     - Underscore (_), hyphen (-)

5. Enter the path to the private part of the SSH key located on the administrator host and that is used for connection to the primary node by KDT (the `key` parameter of the configuration file).The default value is `/root/.ssh/id_rsa`. The parameter value must be a Linux file path or Base64-encoded file content.

6. Enter the number of worker nodes.

   Possible values:

   - 0—Single-node deployment.
   - 3 or more—Multi-node deployment.

This step defines the option of deploying Kaspersky Next XDR Expert. If you want to perform single-node deployment, the following parameters specific for this deployment option will take the default values:

- `type—primary-worker`
- `low_resources—true`
- `openbao_ha_mode—false`
- `openbao_standalone—true`
- `default_class_replica_count—1`

7. For each worker node, enter the IPv4 address (the `host` parameter of the configuration file).

   All nodes must be included in the same subnet as the Kubernetes cluster gateway.

8. For each worker node, enter the user name used for connection to the worker node by KDT (the `user` parameter of the configuration file).

   The value must comply with the following rules:

   - The user name must be 1 to 31 characters long.
   - The user name can contain the following characters:
     - Lowercase letters (a–z)
     - Numbers (0–9)
     - Underscore (_), hyphen (-)

9. For each worker node, enter the path to the private part of the SSH key used for connection to the worker node by KDT (the `key` parameter of the configuration file).The default value is `/root/.ssh/id_rsa`. The parameter value must be a Linux file path or Base64-encoded file content. Also, the value must match the `ssh_pk` parameter value.

10. Enter 1 if you have multiple primary nodes and an external load balancer is required. If you have multiple primary nodes but you do not want to use an external load balancer, enter 0. In this case, worker nodes will act as the load balancer.

11. Enter the IPv4 address of an external load balancer. This value corresponds to the `host` parameter of the configuration file.

12. Enter the user name of the account used for connection to the external load balancer by KDT (the `user` parameter of the configuration file).

The value must comply with the following rules:

- The user name must be 1 to 31 characters long.
- The user name can contain the following characters:
  - Lowercase letters (a–z)
  - Numbers (0–9)
  - Underscore (_), hyphen (-)

13. Enter the path to the private part of the SSH key located on the administrator host and that is used for connection to the external load balancer by KDT (the `key` parameter of the configuration file). The parameter value must be a Linux file path or Base64-encoded file content.

14. Enter the connection string for accessing the DBMS that is installed and configured on a separate server (the `psql_dsn` parameter of the configuration file).

    Specify this parameter as follows: `postgres://<dbms_username>:<password>@<fqdn>:<port>`

    where:

    - `dbms_username`—The user name of a privileged internal DBMS account. This account is granted permissions to create databases and other DBMS accounts. By using this privileged DBMS account, the databases and other DBMS accounts required for the Kaspersky Next XDR Expert components will be created during the deployment.
    - `password`—The password of the privileged internal DBMS account.
    - `fqdn:port`—The FQDN and connection port of the target host on which the DBMS is installed.

      For the single-node deployment, if your DBMS is installed on the same host as a Kubernetes cluster node, do not use `localhost` or `127.0.0.1` as the DBMS address in the connection string. Use the host's FQDN or IP address instead.

    To use a highly available cluster, specify this parameter as follows:
    `psql_dsn=postgres://<dbms_username>:<password>@<fqdn1>:<port>,<fqdn2>:<port>,<fqdn3>:<port>`

    The `psql_dsn` parameter value must comply with the URI format. If the connection URI includes symbols with special meaning in any of its parts, it must be encoded with percent-encoding.

    Symbols that must be replaced in the `psql_dsn` parameter value:

    - Whitespace → `%20`
    - `%` → `%25`
    - `&` → `%26`
    - `/` → `%2F`
    - `:` → `%3A`
    - `=` → `%3D`
    - `?` → `%3F`

- `@` → `%40`
- `[` → `%5B`
- `]` → `%5D`

Refer to the PostgreSQL connection string article for details.

> The Configuration wizard specifies the installation parameters only for the deployment option with the DBMS installed on a separate server that is located outside the Kubernetes cluster.

15. Enter the IPv4 address of the Kubernetes cluster gateway (the `ingress_ip` parameter of the configuration file).

    The gateway must be included in the same subnet as all cluster nodes.

16. Enter the password of the Kaspersky Next XDR Expert user account that will be created by KDT during the installation (the `admin_password` parameter of the configuration file).

    The default user name of this account is "admin." The Main administrator role is assigned to this user account.

    The password must comply with the following rules:

    - The user password cannot have fewer than 8 or more than 256 characters.
    - The password must contain characters from at least three of the groups listed below:
      - Uppercase letters (A–Z)
      - Lowercase letters (a–z)
      - Numbers (0–9)
      - Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)
    - The password must not contain any whitespaces, or the ".@" combination.

    When you specify the `admin_password` parameter value manually (not by the Configuration wizard), make sure that this value meets the YAML standard requirements for values in strings:

    - The parameter value containing special characters must be enclosed in single quotes.
    - Any single quote ' inside the parameter value must be doubled to escape this single quote.

    Example: the user account password `Any_pass%1234'5678"90` must be specified as the value `'Any_pass%1234''5678"90'` of the `admin_password` parameter.

17. Enter the path to the KUMA inventory file located on the administrator host (the `inventory` parameter of the configuration file).

    The KUMA inventory file contains the installation parameters for deployment of the KUMA services that are not included in the Kubernetes cluster. The parameter value must be a Linux file path or Base64-encoded file content.

    If KUMA services are installed in the Kubernetes cluster, specify the path `/dev/null`.

18. Enter the path to the private part of the SSH key located on the administrator host and used for connection to the worker node and nodes with the KUMA services (collectors, correlators, and storages) by using KDT (the `ssh_pk` parameter of the configuration file).

    The parameter value must be a Linux file path or Base64-encoded file content.

19. Enter the path to the LICENSE file of KUMA Core (the `license` parameter of the configuration file).

    The parameter value must be a Linux file path or Base64-encoded file content.

20. Enter the domain name that is used in the FQDNs of the public Kaspersky Next XDR Expert services (the `smp_domain` parameter of the configuration file).

    The parameter value must meet the requirements for second-level domain naming.

21. Enter the path to the custom certificates used to work with the public Kaspersky Next XDR Expert services (the `intermediate_bundle` parameter of the configuration file).

    The parameter value must be a Linux file path or Base64-encoded file content.

    If you want to use self-signed certificates, press **Enter** to skip this step.

22. Skip the step to specify the `extended_incident_lifecycle` parameter. This is a service parameter. By default, the `extended_incident_lifecycle` parameter is disabled, do not change it.

23. Check the specified parameters that are displayed in the numbered list.

    To edit the parameter, enter the parameter number, and then specify a new parameter value. Otherwise, press **Enter** to continue.

24. Press **Y** to save a new configuration file with the specified parameters or **N** to stop the Configuration wizard without saving.

The configuration file with the specified parameters is saved in the YAML format.

Other installation parameters are included in the configuration file, with default values. You can edit the configuration file manually before the deployment of Kaspersky Next XDR Expert.

Installing Kaspersky Next XDR Expert

Kaspersky Next XDR Expert is deployed by using KDT. KDT automatically deploys the Kubernetes cluster within which the Kaspersky Next XDR Expert components and other infrastructure components are installed. The steps of the Kaspersky Next XDR Expert installation process do not depend on the selected deployment option.

> If you need to install multiple Kubernetes clusters with Kaspersky Next XDR Expert instances, you can use the required number of contexts.

To install Kaspersky Next XDR Expert:

1. Unpack the downloaded distribution package with KDT on the administrator host.

2. Read the End User License Agreement (EULA) of KDT located in the distribution package with the Kaspersky Next XDR Expert components.

   When you start using KDT, you accept the terms of the EULA of KDT.

You can read the EULA of KDT after the deployment of Kaspersky Next XDR Expert. The `license_kdt.txt` file is located in the `~/kaspersky_eula` directory of the user who runs the deployment of Kaspersky Next XDR Expert.

3. During installation, KDT downloads missing packages from the OS repositories. Before installing Kaspersky Next XDR Expert, run the following command on the target hosts to make sure that the apt/yum cache is up-to-date.

   ```
   apt update
   ```

4. Restart target hosts.

5. On the administrator host, run the following commands to start deployment of Kaspersky Next XDR Expert by using KDT. Specify the path to the transport archive with the Kaspersky Next XDR Expert components and the path to the configuration file that you filled out earlier (installation parameter sets for the multi-node and single-node deployment differ).

   ```
   chmod +x kdt
   ./kdt apply -k <full_path_to_transport_archive> -i <full_path_to_configuration_file>
   ```

   You can install Kaspersky Next XDR Expert without prompting to read the terms of the EULA and the Privacy Policy of OSMP, if you use the `--accept-eula` flag. In this case, you must read the EULA and the Privacy Policy of OSMP before the deployment of Kaspersky Next XDR Expert. The files are located in the distribution package with the Kaspersky Next XDR Expert components.

   If you want to read and accept the terms of the EULA and the Privacy Policy during the deployment, do not use the `--accept-eula` flag.

   If an error occurs during deployment of Kaspersky Next XDR Expert by using KDT, restart the deployment by specifying the path to the configuration file without the `-i` flag.

6. If you do not use the `--accept-eula` flag in the previous step, read the EULA and the Privacy Policy of OSMP. The text is displayed in the command line window. Press the space bar to view the next text segment. Then, when prompted, enter the following values:

   a. Enter `y` if you understand and accept the terms of the EULA.

      Enter `n` if you do not accept the terms of the EULA.

   b. Enter `y` if you understand and accept the terms of the Privacy Policy, and if you agree that your data will be handled and transmitted (including to third countries) as described in the Privacy Policy.

      Enter `n` if you do not accept the terms of the Privacy Policy.

   > To use Kaspersky Next XDR Expert, you must accept the terms of the EULA and the Privacy Policy.

   After you start the deployment, KDT checks whether the hardware, software, and network configuration of the Kubernetes cluster nodes meet the prerequisites for installing the solution. If all the strict pre-checks are successfully completed, KDT deploys the Kaspersky Next XDR Expert components within the Kubernetes cluster on the target hosts. Otherwise, the deployment will be interrupted. You can skip the pre-checks before the deployment, if needed (set the `ignore_precheck` installation parameter to `true`).

   Also KDT checks whether the required installation parameter are specified correctly. If the required parameters in the configuration file are missing or invalid, the deployment will be interrupted. The rules to be followed by the installation parameter values are specified in the configuration file templates (multinode.smp_param.yaml.template and singlenode.smp_param.yaml.template).

During the Kaspersky Next XDR Expert deployment, a new user is created on the primary Administration Server. To start configuring OSMP Console, this user is assigned the following roles: the XDR role of the Main administrator in the Root tenant and the Kaspersky Security Center role of the Main administrator.

7. View the installation logs of the Bootstrap component in the directory with the KDT utility and obtain diagnostic information about Kaspersky Next XDR Expert components, if needed.

8. Sign in to OSMP Console.

   The OSMP Console address is `https://<console_host>.<smp_domain>:443`.

   Addresses consist of the `console_host` and `smp_domain` parameter values specified in the configuration file.

Kaspersky Next XDR Expert is deployed on the target hosts. Install the KUMA services to get started with the solution.

Configuring internet access for the target hosts

If your organization's infrastructure uses the proxy server to access the internet, as well as you need to connect the target hosts to the internet, you must add the IP address of each target host to the `no_proxy` variable in the /etc/environment file before the Kaspersky Next XDR Expert deployment. This allows you to establish a direct connection of the target hosts to the internet and correctly deploy Kaspersky Next XDR Expert.

To configure internet access for the target hosts:

1. On the target host, open the /etc/environment file by using a text editor. For example, the following command opens the file by using the GNU nano text editor:

   ```
   sudo nano /etc/environment
   ```

2. In the /etc/environment file, add the IP address of the target host to the `no_proxy` variable separated by a comma without a space.

   For example, the `no_proxy` variable can be initially specified as follows:

   ```
   no_proxy=localhost,127.0.0.1
   ```

   You can add the IP address of the target host (192.168.0.1) to the `no_proxy` variable:

   ```
   no_proxy=localhost,127.0.0.1,192.168.0.1
   ```

   Alternatively, you can specify the subnet that includes the target hosts (in CIDR notation):

   ```
   no_proxy=localhost,127.0.0.1,192.168.0.0/24
   ```

3. Save the /etc/environment file.

4. Reboot the target host:

   ```
   reboot
   ```

After you add the IP addresses in the /etc/environment file to each target host and reboot them, you can continue preparing of the target hosts and further Kaspersky Next XDR Expert deployment.

Synchronizing time on machines

To configure time synchronization on machines:

1. Run the following command to install chrony:

   ```
   sudo apt install chrony
   ```

2. Configure the system time to synchronize with the NTP server:

   a. Make sure the virtual machine has internet access.

      If access is available, go to step b.

      If internet access is not available, replace the current public sources (starting with `pool`) in the `/etc/chrony.conf` file with the name or IP address of your organization's internal NTP server.

   b. Start the system time synchronization service by executing the following command:

      ```
      sudo systemctl enable --now chrony
      ```

   c. Wait a few seconds, and then run the following command:

      ```
      sudo timedatectl | grep 'System clock synchronized'
      ```

      If the system time is synchronized correctly, the output will contains the line `System clock synchronized: yes`.

Synchronization is configured.

Configuring deployment from Endpoint Agents via an intermediate proxy node

This section describes a deployment scenario for Kaspersky Next XDR Expert Endpoint Agents and the servers to which they transfer data when located across different networks. This approach simplifies network configuration, while ensuring reliable connectivity between Endpoint Agents and servers.

In this scenario, data from Endpoint Agents is routed through an intermediate proxy node. This allows for a single network connection between the proxy node and the solution, rather than establishing multiple connections from each Endpoint Agent to the solution. This approach can be applied to both single-node and multi-node Kaspersky Next XDR Expert 2.0 deployments.

The section describes setting up an Nginx proxy server on a Linux device. You may use another package that supports mTLS traffic termination and a different operating system if needed.

This scenario uses example domain names (proxy.example.com, backend.example.com) and file and directory paths. You should replace these with values that match your network infrastructure.

Deployment stages

Deployment with configuration of proxying data from Endpoint Agents through an intermediate proxy node involves several stages:

1. **Generating certificates for Nginx**

   Certificates must be created on the server where Nginx will be installed. This server should be on the same network (subnet) as the Endpoint Agents and act as a proxy node.

   1. Prepare the required working directories for certificates:

      ```
      sudo mkdir -p /etc/nginx/mtls && cd /etc/nginx/mtls
      ```

   2. Create a self-signed root certificate (Root CA):

      ```
      openssl genrsa -out ca.key 4096
      openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 \
        -subj "/C=US/O=ExampleCorp/OU=IT/CN=Example Root CA" -out ca.crt
      ```

   3. Create a server certificate for Nginx proxy:

      ```
      openssl genrsa -out asproxy.key 2048
      openssl req -new -key asproxy.key \
        -subj "/C=US/O=ExampleCorp/OU=IT/CN=proxy.example.com" -out asproxy.csr
      openssl x509 -req -in asproxy.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
        -out asproxy.crt -days 825 -sha256 \
        -extfile <(printf
      "subjectAltName=DNS:proxy.example.com\nkeyUsage=digitalSignature,keyEncipherment\nextendedKeyUsage=serverAuth")
      ```

      Server certificate for a proxy node asproxy.crt is created.

   4. Create client certificates for Endpoint Agents (mTLS):

      ```
      openssl genrsa -out client.key 2048
      openssl req -new -key client.key \
        -subj "/C=US/O=ExampleCorp/OU=IT/CN=XDR Client" -out client.csr
      openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial \
        -out client.crt -days 825 -sha256 \
        -extfile <(printf "keyUsage=digitalSignature\nextendedKeyUsage=clientAuth")
      ```

      The following client keys and certificates for Endpoint Agents are created:

      - client.key—PEM, private key
      - client.csr—PEM, certificate signing request
      - client.crt—PEM, client certificate

5. Convert certificate to PKCS#12 format:

```
openssl pkcs12 -export -out client.p12 -inkey client.key -in client.crt -certfile ca.crt -name "EDR client cert"
```

When prompted, enter a strong certificate password.

As a result of this step, a client certificate in the PKCS#12 format `client.p12` is obtained.

6. Set access rights to keys:

```
chmod 600 ca.key asproxy.key client.key
chown root:root ca.* asproxy.* client.*
```

2 Installing Nginx

Install Nginx proxy server on a server located in the same network (subnet) where the Endpoint Agents are located.

1. Run the following commands:

```
sudo apt update
sudo apt install nginx
```

2. Check for the presence of the http_ssl_module module (included in Nginx by default):

```
nginx -V
```

The command output should contain the http_ssl_module module.

Nginx proxy server is installed.

3 Configuring Nginx as an mTLS proxy

1. Create or edit the existing Nginx configuration file using the template.

Default Nginx configuration file path: /etc/nginx/conf.d/asproxy.conf.

Configuration file template

```
# Map for WebSocket support
map $http_upgrade $connection_upgrade {
  default upgrade;
  ''      close;
}

server {
  listen 443 ssl http2;
  listen 9443 ssl;
  server_name proxy.example.com;

  # mTLS at edge configuration
  ssl_certificate         /etc/nginx/mtls/asproxy.crt;
  ssl_certificate_key     /etc/nginx/mtls/asproxy.key;
  ssl_trusted_certificate /etc/nginx/mtls/ca.crt;
  ssl_client_certificate  /etc/nginx/mtls/ca.crt;
  ssl_verify_client       on;
  ssl_verify_depth        2;
  ssl_protocols           TLSv1.2 TLSv1.3;
  ssl_session_cache       shared:SSL:10m;

  # Forward client identity to upstream
```

```
    proxy_set_header X-Client-Verify $ssl_client_verify;
    proxy_set_header X-Client-DN     $ssl_client_s_dn;
    proxy_set_header X-Client-Cert   $ssl_client_escaped_cert;

    # Common proxy headers
    proxy_http_version                1.1;
    proxy_set_header Host             backend.example.com;
    proxy_set_header X-Real-IP        $remote_addr;
    proxy_set_header X-Forwarded-For  $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;

    proxy_connect_timeout 5s;
    proxy_send_timeout    60s;
    proxy_read_timeout    60s;
    send_timeout          60s;

    proxy_ssl_server_name on;
    proxy_ssl_name        backend.example.com;
    proxy_ssl_verify      off;   # Verification disabled for internal backend
    proxy_ssl_protocols   TLSv1.2 TLSv1.3;

    # Common HTTPS
    location / {
      proxy_pass https://backend.example.com;
    }

    # WSS for LiveShell
    location ^~ /vthost/sessions {
      proxy_pass https://backend.example.com:9443;

      proxy_set_header Upgrade    $http_upgrade;
      proxy_set_header Connection $connection_upgrade;

      proxy_read_timeout  3600s;
      proxy_send_timeout  3600s;

      proxy_request_buffering off;
      proxy_buffering         off;
    }

    access_log /var/log/nginx/proxy.example.com.access.log;
    error_log  /var/log/nginx/proxy.example.com.error.log warn;

    error_page 495 496 497 = /__mtls_error;
    location = /__mtls_error { return 403; }
  }
```

Template variables:

- proxy.example.com—FQDN of a proxy node with Nginx
- backend.example.com—FQDN of a target Kaspersky Next XDR Expert 2.0 backend server (Agentserver in KEDR 7.1).

2. Restart Nginx and apply the created configuration file:

```
nginx -s reload
```

Installed Nginx proxy server is configured for Endpoint Agent data proxying.

4 Configuring Kaspersky Endpoint Security for Windows Endpoint Agents to work through a proxy node

1. Configure network connectivity. Ensure that network settings (firewall, routing) allow outgoing connections from the proxy node to the Kaspersky Next XDR Expert platform over the TCP:443 and TCP:8443 protocols.

2. Copy client certificate files created in Step 1.4 of this procedure to the server where Endpoint Agent is installed (C:\Users\TestUser\ in the example in this section).

3. On the host where Endpoint Agent is installed (for example, Kaspersky Endpoint Security for Windows), navigate to the directory containing the executable file and run the following command:

```
avp.com EDREXPERTONPREM `
/SET /mode=EDRExpertOnPrem `
/servers=proxy.example.com:443 `
/server-certificate=C:\Users\TestUser\asproxy.crt `
/client-certificate=C:\Users\TestUser\client.p12 `
/client-certificate-password=YourStrongPassword `
/telemetry-servers=telemetry.example.com:443 `
/sync-period=1
```

For more information on configuring an Endpoint Agent using the command line, see the Kaspersky Endpoint Security documentation.

Installing KUMA services inside the Kubernetes cluster

Services are the main components of the KUMA SIEM-system, which helps the system to manage events. Services allow you to receive events from event sources and subsequently bring them to a common form that is convenient for finding correlation, as well as for storage and manual analysis.

Service types:

- Storages are used to save events.
- Collectors are used to receive events and convert them to the KUMA format.

  If you plan to use multiple tenants in your infrastructure, create a separate collector for each tenant.

- Correlators are used to analyze events and search for defined patterns.
- Agents are used to receive events on remote devices and forward them to the KUMA collectors.

You must install the KUMA services only after you deploy Kaspersky Next XDR Expert. During the Kaspersky Next XDR Expert deployment, the required infrastructure is prepared: the service directories are created on the prepared hosts, and the files that are required for the service installation are added to these directories. We recommend installing services in the following order: storage, collectors, correlators, and agents.

Limitations

When installing KUMA services inside the Kubernetes cluster, consider the following limitations:

- Only one instance of KUMA Core can be deployed per one installation.
- Passive collectors can be deployed only when HTTP or TCP connectors with TLS enabled are used.

- For passive HTTP and TCP connectors, the configured server address and port are replaced automatically with `0.0.0.0:7310`.
- A storage cluster must be deployed entirely inside Kubernetes or entirely outside it. Mixed deployments are not supported.
- If collectors or correlators deployed outside Kubernetes send events to storage services deployed inside Kubernetes, event delivery is performed through a single ingress URL. Load balancing between storage nodes is performed by Kubernetes at the TCP level.
- If traffic must be routed to different storage node groups, you need to deploy separate eventRouter services and route traffic through them.
- When you change settings of a collector deployed inside the Kubernetes cluster, KUMA does not verify the modified settings. Incorrect settings may prevent the collector from receiving events.
- Multiple storage services created from the same storage resource cannot be deployed in a single Kubernetes cluster using the same **Storage Node FQDN** value.

To install and configure the KUMA services inside the Kubernetes cluster:

1. Ensure that the administrator and target hosts for Kaspersky Next XDR Expert are prepared according to the selected deployment scheme by following one of the instructions below:
   - Multi-node deployment: Preparing the administrator and target hosts
   - Single-node deployment: Preparing the administrator and target hosts
2. In the OSMP Console, create a resource set for each KUMA service (storages, collectors, and correlators).
3. Create services for storages, collectors, and correlators.
4. **Deploy each service into the Kubernetes cluster**.

   1. In the main menu, go to **Monitoring** → **Resources and services** → **Active services**.
   2. Select the service you want to deploy and click the **Deploy** button.

3. In the **Add service to Kubernetes** window, fill in the following fields:

   - **Storage Node FQDN**: Specify the name of the storage cluster node defined in the storage resource. This parameter is available only for a storage service and cannot be changed after the service is deployed.

   - **Storage Class**:Select the storage subsystem used to store the service state. The value of this parameter cannot be changed after deployment. Available values:

     - **longhorn**—this storage class provides network storage with replication support. By default, the service state is replicated across three network drives. This storage class is recommended for collector, eventRouter, correlator, metrics, and Core services because it allows Kubernetes to migrate the service pod to another worker node if the current node fails. It can also be used to provide high availability for a storage service deployed as a single instance.

     - **local-path**—this storage class uses local disk arrays of Kubernetes worker nodes mounted under /opt/lvp/hot directory and provides maximum performance. This storage class is recommended for a storage service because storage supports native clustering and does not rely on Kubernetes mechanisms for high availability. It can also be used for collector services that use HTTP or TCP connectors and for eventRouter services deployed in multiple instances created from a single KUMA resource. In single-node deployment, using **local-path** is recommended for all KUMA service types.

   - **Cold Storage Class**: Select the storage subsystem used for long-term data storage. This parameter is available only for a storage service and cannot be changed after deployment. The **local-path-cold** storage class is similar to **local-path** but is intended for long-term storage on slower disk arrays mounted under /opt/lvp/cold directory.

   - **Minimum CPU**: Specifiy the number of CPU cores (threads) reserved for the service. The value must be a positive integer. The default value is `0`.

   - **Maximum CPU**: Specify the maximum number of CPU cores (threads) available to the service. The value must be a positive integer and cannot be less than the **Minimum CPU** value. The default value is `0`.

   - **Minimum RAM**: Specifiy the amount of RAM in GB, reserved for the service. The value must be a positive integer. The default value is `0`.

   - **Maximum RAM**: Specify the maximum amount of RAM in GB, available to the service. The value must be a positive integer and cannot be less than the **Minimum RAM** value. The default value is `0`.

   - **Storage request**: Specify the amount of disk space in GB, allocated to the service. This parameter is available only when the **longhorn** storage class is used.

   - **Cold Storage request**: Specify the amount of disk space in GB, allocated for long-term data storage. This parameter is available only for storage services that use the **longhorn** storage class.

4. Click the **Create service** button.

5. If necessary, verify that the collector and correlator are ready to receive events.

6. If necessary, install agents in the KUMA network infrastructure.

   The files required for the agent installation are located in the `/opt/kaspersky/kuma/utils` directory.

The KUMA services required for the function of Kaspersky Next XDR Expert are installed inside the Kubernetes cluster.

Installing KUMA services outside the Kubernetes cluster

Services are the main components of the KUMA component, which helps the system to manage events. Services allow you to receive events from event sources and subsequently bring them to a common form that is convenient for finding correlation, as well as for storage and manual analysis.

The KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on worker nodes, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster.

Service types:

- Storages are used to save events.
- Collectors are used to receive events and convert them to the KUMA format.

  > If you plan to use multiple tenants in your infrastructure, create a separate collector for each tenant.

- Correlators are used to analyze events and search for defined patterns.
- Agents are used to receive events on remote devices and forward them to the KUMA collectors.

You must install the KUMA services only after you deploy Kaspersky Next XDR Expert. During the Kaspersky Next XDR Expert deployment, the required infrastructure is prepared: the service directories are created on the prepared hosts, and the files that are required for the service installation are added to these directories. We recommend installing services in the following order: storage, collectors, correlators, and agents.

To install and configure the KUMA services outside the Kubernetes cluster:

1. In the main menu of OSMP Console, go to **Settings** → **KUMA**.
2. In KUMA section of OSMP Console, create a resource set for each KUMA service (storages, collectors, and correlators) that you want to install on the prepared hosts in the network infrastructure.
3. Create services for storages, collectors, and correlators.
4. Obtain the service identifiers to bind the created resource sets and the KUMA services:
   a. In the KUMA section of OSMP Console, go to **Resources** → **Active services**.
   b. Select the required KUMA service, and then click the **Copy ID** button.
5. On the prepared hosts in the network infrastructure, run the corresponding commands to install the KUMA services. Use the service identifiers that were obtained earlier:
   - Installation command for the storage:

     ```
     sudo /opt/kaspersky/kuma/kuma storage --core https://<KUMA Core server FQDN>:7210 --
     id <service ID copied from the KUMA section> --install
     ```

- Installation command for the collector:

  ```
  sudo /opt/kaspersky/kuma/kuma collector --core https://<KUMA Core server FQDN>:7210
  --id <service ID copied from the KUMA section> --api.port <port used for
  communication with the collector>
  ```

- Installation command for the correlator:

  ```
  sudo /opt/kaspersky/kuma/kuma correlator --core https://<KUMA Core server FQDN>:7210
  --id <service ID copied from the KUMA section> --api.port <port used for
  communication with the correlator> --install
  ```

By default, the FQDN of KUMA Core is `<kuma_host>.<smp_domain>`.

The port that is used for connection to KUMA Core cannot be changed. By default, port 7210 is used.

Open ports that correspond to the installed collector and correlator on the server (TCP 7221 and other ports used for service installation as the `--api.port <port>` parameter values).

6. During the installation of the KUMA services, read the End User License Agreement (EULA) of KUMA. The text is displayed in the command line window. Press the space bar to view the next text segment. Then, when prompted, enter the following values:

   - Enter `y` if you understand and accept the terms of the EULA.

   - Enter `n` if you do not accept the terms of the EULA. To use the KUMA services, you must accept the terms of the EULA.

   You can read the EULA of KUMA after the installation of the KUMA services in one of the following ways:

   - On hosts, it is included in the `kuma_utils` group in the KUMA inventory file: open the LICENSE file located in the `/opt/kaspersky/kuma/utils` directory.

   - On hosts, it is included in other groups (`kuma_storage`, `kuma_collector`, or `kuma_correlator`) in the KUMA inventory file: open the LICENSE file located in the `/opt/kaspersky/kuma` directory.

   - Run the following command:

     ```
     /opt/kaspersky/kuma/kuma license --show
     ```

   After you accept the EULA, the KUMA services are installed on the prepared machines in the network infrastructure.

7. If necessary, verify that the collector and correlator are ready to receive events.

8. If necessary, install agents in the KUMA network infrastructure.

   The files required for the agent installation are located in the `/opt/kaspersky/kuma/utils` directory.

The KUMA services required for the function of Kaspersky Next XDR Expert are installed.

Deployment of multiple Kubernetes clusters and Kaspersky Next XDR Expert instances

KDT allows you to deploy multiple Kubernetes clusters with Kaspersky Next XDR Expert instances and switch between them by using contexts. *Context* is a set of access parameters that define the Kubernetes cluster that the user can select to interact with. The context also includes data for connecting to the cluster by using KDT.

Prerequisites

Before creating contexts and installing Kubernetes clusters with Kaspersky Next XDR Expert instances, you must do the following:

1. Prepare the administrator and target hosts.

   For the installation of multiple clusters and Kaspersky Next XDR Expert instances, you need to prepare one administration host for all clusters and separate sets of target hosts for each of the clusters. Kubernetes components should not be installed on the target hosts.

2. Prepare the hosts for installation of the KUMA services.

   For installation of the KUMA services, you need to prepare separate sets of hosts for each Kaspersky Next XDR Expert instance.

3. Prepare the KUMA inventory file.

   For installation of the KUMA services, you need to prepare separate inventory files for each Kaspersky Next XDR Expert instance.

4. Prepare the configuration file.

   For installation of multiple clusters and Kaspersky Next XDR Expert instances, you need to prepare configuration files for each Kaspersky Next XDR Expert instance. In these configuration files, specify the corresponding administration and target hosts, and other parameters specific to a particular cluster and Kaspersky Next XDR Expert instance.

Process

To create a context with the Kubernetes cluster and Kaspersky Next XDR Expert instance:

1. On the administrator host where the KDT utility is located, run the following command and specify the context name:

   ```
   ./kdt ctx --create <context_name>
   ```

   The context with the specified name is created.

2. Install the Kubernetes cluster and Kaspersky Next XDR Expert.

The cluster with the Kaspersky Next XDR Expert instance is deployed in the context. The creation of the context is finished. When you obtain log files of Kaspersky Next XDR Expert components, the log files contain your current context name.

You can repeat this procedure to create the required number of contexts with installed clusters and Kaspersky Next XDR Expert instances.

> You must deploy the Kubernetes cluster and the Kaspersky Next XDR Expert instance after you create the context to finish the context creation. If you do not perform the deployment in the context, and then create another context, the first context will be removed.

To view the list of created contexts and the active context name,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt ctx
```

To switch to the required context,

On the administrator host where the KDT utility is located, run the following command and specify the context name:

```
./kdt ctx <context_name>
```

After you select the context, KDT connects to the corresponding Kubernetes cluster. Now, you can work with this cluster and the Kaspersky Next XDR Expert instance. KDT commands are applied to the selected cluster.

When you remove the Kaspersky Next XDR Expert components installed in the Kubernetes cluster and the cluster itself by using KDT, the corresponding contexts are also removed. Other contexts and their clusters with Kaspersky Next XDR Expert instances are not removed.

Installing and upgrading Kaspersky Next XDR Expert in an air-gapped environment

Kaspersky Next XDR Expert can be installed or updated in an air-gapped environment. Installation and update in an air-gapped environment is performed by using the kdt-airgap-preparation utility, and consists of the following stages:

1. Preparing the administrator and target hosts, and the configuration file.
2. Obtaining information about packages installed on the target hosts located in the air-gapped environment.

   This information is compared with the list of required packages and saves information about what is missing on target hosts to the YAML inventory file.
3. Downloading the required packages specified in the inventory file and their dependencies from public internet repositories or corporate mirror repositories in the open environment.

   The downloaded packages are saved into a local TAR package archive.
4. Installing or updating packages and their dependencies on the target hosts in the air-gapped environment from the local TAR archive.
5. Installation or upgrade of Kaspersky Next XDR Expert in the air-gapped environment.

Preparing the administrator and target hosts

Based on the selected deployment scheme, you must define the number of target hosts on which you will deploy the Kubernetes cluster and the Kaspersky Next XDR Expert components included in this cluster. Prepare the administrator host and target hosts for deployment of Kaspersky Next XDR Expert by using one of the following instructions, except steps with downloading and installation of packages:

- Multi-node deployment: Preparing the administrator and target hosts
- Single-node deployment: Preparing the administrator and target hosts

Prepare the configuration file in the YAML format. The configuration file contains the list of target hosts for deployment and a set of installation parameters of the Kaspersky Next XDR Expert components.

Use the configuration file that contains the installation parameters specific for the selected deployment scheme:

- Multi-node deployment: Specifying the installation parameters
- Single-node deployment: Specifying the installation parameters

Obtaining information about packages installed on the target hosts located in the air-gapped environment

To obtain information about the operating system and installed packages of target hosts:

1. On the administrator host, run the following command. Specify the path to the private part of the SSH key located on the administrator host and that is used for connection to target hosts, the path to the configuration file that you filled out earlier, and path to the inventory file to which information will be saved:

   ```
   kdt-airgap-preparation scan -k <full_path_to_ssh_key> -c
   <full_path_to_configuration_file> --airgap -i <full_path_to_inventory_file>
   ```

   The kdt-airgap-preparation utility compares the information with the list of required packages and saves information about what is missing on target hosts to the YAML inventory file. The YAML inventory file is saved to the specified directory, with the specified name.

2. If you need to check for additional packages on the target hosts, such as Docker, run the following command:

   ```
   kdt-airgap-preparation scan -k <full_path_to_ssh_key> -c
   <full_path_to_configuration_file> --airgap -i <full_path_to_inventory_file> --add-
   pkgs <package_name>
   ```

   Information about additional packages that are not installed on the target hosts is added to the YAML inventory file.

The created inventory file will be used to download missing packages for target hosts from public internet repositories or corporate mirror repositories in the open environment. If an error occurs, the kdt-airgap-preparation utility terminates and outputs the corresponding error message to both the log file and stdout.

Downloading the required packages in the open environment

To download the required packages specified in the inventory file:

1. Copy the YAML inventory file to a removable drive, and then move the file to the administrator host in the open environment.

2. On the administrator host, run the following command. Specify the path to the inventory file and path to the package archive to which the required missing packages should be saved:

   ```
   kdt-airgap-preparation pull -n <full_path_to_inventory_file> -a
   <full_path_to_package_archive>
   ```

The kdt-airgap-preparation utility downloads the packages listed in the inventory file and their dependencies and saves packages to the TAR package archive. The package archive is saved to the specified directory, with the specified name.

3. Add the inventory file to the package archive.

   The created package archive will be used to install the missing required packages on target hosts in the air-gapped environment. If an error occurs, the kdt-airgap-preparation utility terminates and outputs the corresponding error message to both the log file and stdout.

Installing or updating packages on the target hosts in the air-gapped environment

To install or update packages and their dependencies on the target hosts:

1. Copy the TAR package archive to a removable drive, and then move the archive to the administrator host in the air-gapped environment.

2. On the administrator host, run the following command. Specify the path to the TAR package archive and the path to the private part of the SSH key located on the administrator host and used for connection to target hosts:

   ```
   kdt-airgap-preparation install -a <full_path_to_package_archive> -k
   <full_path_to_ssh_key>
   ```

   The kdt-airgap-preparation utility installs or updates the required packages on the target hosts by using the inventory file and the package archive.

After you prepare target hosts, install or upgrade Kaspersky Next XDR Expert.

Pre-check of infrastructure readiness for deployment

After you start the deployment of Kaspersky Next XDR Expert, KDT checks whether the hardware, software, and network configuration of the Kubernetes cluster nodes meet the prerequisites for installing the solution. A pre-check is performed for each node of the cluster.

All checks are divided into two groups:

- Strict checks

  Checking the parameters that are critical for the operation of Kaspersky Next XDR Expert. If this check fails, the deployment is interrupted.

- Non-strict checks

  Checking the parameters that are not critical for the operation of Kaspersky Next XDR Expert. If this check fails, the deployment continues.

The following pre-checks are performed:

- Hardware:

  - Free space on disks is enough for deployment.

  - CPU configuration meets the requirements.

    - Free RAM space is enough for deployment.
    - CPU supports the AVX, SSE2, and BMI instructions.
    - The amount of free disk space in the `/tmp` directory must be at least equal to the size of the transport archive (for administrator host, strict check).
- Software:
    - Operating system and its version meet the requirements.
    - Kernel version meets the requirements.
    - Systemctl is installed and available (strict check).
    - Update of the package manager cache is available (strict check).
    - Required packages of the correct version are installed on the node (strict check).
    - Prohibited packages (docker and podman) are not installed on the node (strict check).
    - Outdated k0s binaries are missing (strict check).
    - Outdated k0s configuration files are missing (strict check).
    - PostgreSQL or Postgres Pro DBMS server parameters meet the requirements (strict check).
- Network:
    - All cluster nodes are located in the same broadcast domain.
    - All cluster nodes have unique names (the `desc` parameter of the configuration file) (strict check).
    - Custom certificate file is valid for the specified domain and contains the private encrypted key (strict check).
    - The validity period for the custom intermediate certificate is 399 days or more (strict check).
    - The validity period for the custom leaf certificate is 399 days or more (strict check).
    - DNS name resolution on the node is available (strict check).
    - Time synchronization on cluster nodes is configured (strict check).
    - The local time on all nodes differs by no more than 5 seconds (strict check).
    - Required ports are available.

If all the strict pre-checks are successfully completed, KDT deploys the Kaspersky Next XDR Expert components within the Kubernetes cluster on the target hosts. The results of all passed and failed checks are saved on each node in the file /tmp/k0s_report.txt. You can skip the pre-checks before the deployment, if needed (set the `ignore_precheck` installation parameter to `true`).

Signing in to Kaspersky Next XDR Expert

To sign in to Kaspersky Next XDR Expert, you must know the web address of Open Single Management Platform Console. In your browser, JavaScript must be enabled.

To sign in to Open Single Management Platform Console:

1. In your browser, go to `https://<console_host>.<smp_domain>:443`.

   The Open Single Management Platform Console address consists of the `console_host` and `smp_domain` parameter values specified in the configuration file.

   The sign-in page is displayed.

2. Do one of the following:

   - To sign in to Open Single Management Platform Console by using a domain user account, enter the user name and password of the domain user.

     You can enter the user name of the domain user in one of the following formats:

     - `Username@dns.domain`
     - NTDOMAIN\`Username`

     Before you sign in with a domain user account, poll the domain controller to obtain the list of domain users.

   - Enter the user name and password of the internal user.
   - If one or more virtual Servers are created on the Server and you want to sign in to a virtual Server:

     a. Click **Show virtual Server options**.

     b. Type the virtual Server name that you specified while creating the virtual Server.

     c. Enter the user name and password of the internal or domain user who has rights on the virtual Server.

3. Click the **Sign in** button.

4. Specify the one-time security code generated by the authenticator app on the mobile device.

   At the first sign-in, configure two-step authentication for your account. If two-factor authentication configuration is not available to you, contact a user who has the Modify object ACLs right in the **General features: User permissions** functional area and configured two-factor authentication to add your account to the two-factor authentication allowlist.

After sign-in, the dashboard is displayed, and it contains the language and theme that you used the last time you signed in.

Kaspersky Next XDR Expert allows you to work with Open Single Management Platform Console and KUMA Console interfaces.

If you sign in to one of the consoles, and then open the other console on a different tab of the same browser window, you are signed in to the other console without having to re-enter the credentials. In this case, when you sign out of one console, the session also ends for the other console.

If you use different browser windows or different devices to sign in to Open Single Management Platform Console and KUMA Console, you have to re-enter the credentials. In this case, when you sign out of one console on the browser window or device where it is open, the session continues on the window or device where the other console is open.

To sign out of Open Single Management Platform Console,

In the upper pane, go to your account menu by clicking your account name, and then select **Sign out**.

Open Single Management Platform Console is closed and the sign-in page is displayed.

Integration with Endpoint Agents

To integrate an Endpoint Agent, you need to get certificates required for establishing a *trusted connection* between OSMP and an Endpoint Agent.

To get OSMP certificate for receiving telemetry from Endpoint Agents:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Active services**.
2. In the list of services, right-click on the collector that you want to use for collecting telemetry sent by Endpoint Agents, and then select **Download certificate** from the context menu.

   Your browser downloads the server TLS certificate to the downloads folder.

To get OSMP certificate for sending responses to Endpoint Agents:

1. In the main menu, go to **Settings** → **Tenants** → Click on the tenant name → **Settings** tab → **Certificates**.
2. in the **Server** section, click **Export**.

   Your browser downloads the server TLS certificate to the downloads folder.

To get Endpoint Agent TLS certificate:

1. In the main menu, go to **Settings** → **Tenants** → Click on the tenant name → **Settings** tab → **Certificates**.
2. In the **Endpoint Agents** section, click **Generate**.
3. Specify a password for a password-protected PFX archive containing the Endpoint Agent TLS certificate, and then click **Export**.

   Your browser downloads the PFX archive (crypto container) to the downloads folder.

   The obtained certificate is required for mutual TLS authentication when sending responses to Endpoint Agents and receiving telemetry from them.

After you have downloaded all required certificates, you can create a policy for integrating an Endpoint Agent. For detailed instructions on how to create a policy and add certificates, refer to an Endpoint Agent documentation:

- Kaspersky Endpoint Security for Windows
- Kaspersky Endpoint Security for Linux
- Kaspersky Endpoint Security for Mac

Sandbox preparation and installation

To prepare for installation and install Sandbox:

1. Prepare your IT infrastructure for the Sandbox installation.
2. Prepare a virtual machine.
3. Prepare an installation disk image.
4. Install Sandbox.

Preparing IT infrastructure for the Sandbox installation

To prepare your IT infrastructure for the Sandbox installation:

1. Deny access to the local network of the Sandbox server for the management network interface and the network interface used for internet access of processed objects.
2. Specify the ports for interaction with other components as follows:
   - Inbound direction (management interface):
     - TCP 22
       Connecting to the server over SSH
     - TCP 443
       Communication with OSMP
     - TCP 8443
       Access to the web interface of the application
   - Outbound direction (management interface):
     - TCP 80 or TCP 443
       Communication with Kaspersky update servers.
   - Outbound and corresponding inbound directions (interface for access of processed objects):
     Any port.

Access to the internet for analyzing the network behavior of processed objects.

Deny access to the corporate LAN to protect the network from analyzed objects.

Preparing a virtual machine for installing Sandbox

To prepare the virtual machine for installing the Sandbox component:

1. Run the VMware ESXi hypervisor.
2. Open the virtual machine management console.
3. In the context menu of the virtual machine on which you want to install the Sandbox component, choose **Edit Settings**.

   Virtual machine properties window opens.
4. On the **Virtual Hardware** tab, expand the CPU settings group and select the **Expose hardware-assisted virtualization to guest OS** check box.
5. On the **VM Options** tab in the **Latency Sensitivity** drop-down list, select **High**.
6. Click **OK**.

The virtual machine is ready for installing the Sandbox component.

Preparing an installation disk image

Before installing Sandbox, you must prepare an iso image of the installation disk based on the Astra Linux operating system.

Minimum hardware requirements for a device that can be used to create the iso image:

- CPU: 4 cores, clock rate 2500 MHz or more.
- RAM: 8 GB.
- Available disk space: 100 GB.

Software requirements:

- Operating system based on an up-to-date Linux kernel.
- Docker 20 or later
- An ISO image of Astra Linux Special Edition 1.8.2.8 (hereinafter this image is referred to as astra-installation.iso).

Kaspersky Next XDR Expert does not support other versions of the Astra Linux operating system.

To mount an iso image of the Sandbox component based on the Astra Linux operating system:

1. Download the following files from the distribution kit:

   - tar.gz archive from the Sandbox distribution kit
   - the tool for creating an ISO image

2. Create an iso_builder.sh file with the following content:

```
#!/bin/bash

usage="Builds the KEDR install ISO.

Usage:

$(basename "$0") [-h] source_iso distribution iso_builder_image build target_iso_name

where:
    -h, --help        - show this text and exit
    source_iso        - absolute path to the source OS ISO file
    distribution      - absolute path to the KEDR distribution archive
    iso_builder_image - absolute path to the iso_builder docker image
    build             - absolute path to a temporary build directory
    target_iso_name   - absolute path to the resulting ISO file
"

if [ "$#" -eq 0 ] || [ "$1" == "-h" ] || [ "$1" == "--help" ]
then
    echo "$usage"; exit 10;
fi

if [ "$#" -lt 5 ]
then
    echo "Error: not enough arguments. Please, refer the help text."; exit 11;
fi

docker load -i $3
docker run -v $1:$1 -v $2:$2 -v $4:/build kaspersky/kata/deployment/iso_builder:6.0 --source-iso-uri file://$1
--kata-distribution-uri file://$2 --target-iso-name $5
```

3. Run the following command:

```
mkdir /var/kedr_builder
```

4. Put the files listed in step 1 and the astra-installation.iso image in the newly created /var/kedr_builder directory.

5. Run the following command:

```
sudo ./iso_builder.sh \
 /var/kedr_builder/astra-installation.iso \
 /var/kedr_builder/kata-sb-distribution-<version>-x86_64_en-ru-zh.tar.gz \
 /var/kedr_builder/iso-builder-<version>-x86_64_en-ru-zh.tar \
 /var/kedr_builder buildSandboxAstra.iso
```

   where <version> is the Sandbox distribution version.

   The installation disk image buildSandboxAstra.iso is created in the /var/kedr_builder directory.

If you keep files in different directories, you can run the following command:

```
sudo ./iso_builder.sh <source_iso_host_path> <distribution_host_path> <iso_builder_image_host_path>
<build_host_path> <target_iso_name>
```

In the above command:

- source_iso_host_path is the path to the distribution kit of Astra Linux Special Edition 1.8.2.8.
- distribution_host_path is the path to the Sandbox distribution kit.
- iso_builder_image_host_path is the path to the ISO builder tool.
- build_host_path is the path where the mounted ISO image must be located, without specifying the name being assigned to the ISO image.
- target_iso_name is the name assigned to the iso image.

Installing Sandbox

To begin the installation of the Sandbox component:

1. Run the disk image containing the Sandbox component.

   The Setup Wizard starts.

2. Select **Continue** and press **ENTER**.

Step 1. Viewing the End User License Agreement and Privacy Policy

To proceed with the installation, you must read and accept the terms and conditions of the End User License Agreement and the Privacy Policy of AO Kaspersky Lab (hereinafter also referred to as the "EULA" and the "Privacy Policy" respectively). Installation will not continue until you accept the terms of EULA and Privacy Policy.

On the Astra Linux operating system, you also need to accept the End User License Agreement of the Astra Linux operating system.

To accept the terms and conditions of EULA and Privacy Policy:

1. Select the language for reading the EULA and Privacy Policy texts in the list and press the **Enter** key.
2. On the Astra Linux operating system, you must first accept the End User License Agreement of the Astra Linux operating system. Read the EULA. If you accept its terms and conditions, select **I accept** and press **Enter**.
3. Read the EULA of Kaspersky. If you accept its terms and conditions, select **I accept** and press **Enter**.
4. Read the Privacy Policy. If you accept its terms and conditions, select **I accept** and press **Enter**.

The Setup Wizard proceeds to the next step.

Step 2. Selecting a disk for installing Sandbox

To select a disk for installing the component:

1. Select one of the suggested drives for installing the component and press **`Enter`**.
2. In the confirmation window, select **Yes** and press **`Enter`**.

The Setup Wizard proceeds to the next step.

Step 3. Configuring the minimum password length for the Sandbox administrator password

To set the minimum length of the administrator password for the Sandbox component:

1. In the **Minimal password length**, enter the length in characters.
   Passwords 12 or more characters long are recommended.
2. Select **OK** and press **`Enter`**.

The Setup Wizard proceeds to the next step.

Step 4. Creating the Sandbox administrator account

To create a Sandbox administrator account:

1. In the **New username** field, enter the name of the administrator account. The 'admin' account is used by default.
2. In the **New password** field, enter the password for the administrator account.
   The password must comply with the following rules:
   - The password must contain at least 8 characters.
   - The password must contain at least three types of characters:
     - Uppercase character (A-Z).
     - Lowercase character (a-z).
     - Number.
     - Special character.
   - The password must not be completely or substantially the same as the user name.
   - The password may not contain three or more identical characters in a row.

3. Enter the password again in the **Confirm password** field.

4. Select **OK** and press **`Enter`**.

The Setup Wizard proceeds to the next step.

Step 5. Configuring the management network interface

To ensure proper functioning of the Sandbox component, you must connect at least two network cards and configure the following network Interfaces:

- Management network interface. This interface is intended for providing SSH access to the Sandbox server; the Sandbox server also uses this interface to receive objects for scanning.

- Network interface used for Internet access of processed objects. Objects that are processed by Sandbox can use this interface to attempt activities on the Internet, and Sandbox can analyze their behavior. If you block Internet access, Sandbox cannot analyze the behavior of objects on the Internet, and will therefore only analyze the behavior of objects without internet access.

  The network interface used for Internet access of processed objects must be isolated from the local network of your organization.

Select the network interface that you want to use as the management interface and configure it.

To select a management network interface and configure it:

1. In the list of network interfaces, select the network interface that you want to use as the management interface.

2. Press **`Enter`**.

3. In the **Address** field, enter the IP address that you want to assign to this network interface.

4. In the **Netmask** field, enter the network mask in which you want to use this network interface.

5. Select **OK** and press **`Enter`**.

6. Select **Continue** and press **`Enter`**.

The Setup Wizard proceeds to the next step.

Step 6. Assigning a hostname and adding DNS server addresses

Assign a server host name to be used by DNS servers and add DNS server addresses.

To assign a server host name and add DNS server addresses:

1. In the **DNS settings** window, select **Host name**.

2. Enter the full domain name of the server into the **Host name** field.

   Specify the server name in FQDN format (for example: host.domain.com or host.domain.subdomain.com).

3. Select **OK** and press **`Enter`**.

4. In the **DNS settings** window, select **DNS servers**.

5. Select **New** and press **`Enter`**.

   The DNS server address entry window opens.

6. In the **DNS server** text box, enter the IP address of the primary DNS server in IPv4 format.

7. Select **OK** and press **`Enter`**.

   The DNS server address entry window closes.

8. If you want to add the IP address of an additional DNS server, repeat steps 5 through 7 of these instructions for each server that you want to add.

9. Having added all DNS servers, select **Go back** and press **`Enter`**.

10. In the DNS servers window, select **Continue** and press **`Enter`**.

The Setup Wizard proceeds to the next step.

Step 7. Configuring a static network route

To configure a static network route:

1. In the **IPv4 Routes** window, select **New** and press **`Enter`**.

   The **IPv4 Static Route** window opens .

2. In the **Address/Mask** field, enter the IP address and mask of the subnet for which you want to configure the network route.

3. If you want to use the default network route, enter **`0.0.0.0/0`**.

4. In the **Gateway** field, enter the IP address of the gateway.

5. Select **OK** and press **`Enter`**.

6. If you want to add other network routes, repeat steps 1 through 5 of these instructions for each network route that you want to add.

7. If you are done adding network routes, click **Continue**.

The configuration may take some time. Then the installation is complete.

Integration with Sandbox

Integration with Sandbox involves two steps:

1. Creating a request to connect to the Sandbox server in the OSMP interface.
2. Accepting the request in the Sandbox web interface.

To create a connection request to a Sandbox server through the OSMP interface:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. In the **Settings** tab, go to the **Sandbox servers** section.

   Note the *OSMP certificate fingerprint*. You will need this fingerprint later to match with the one displayed in the Sandbox web interface.

4. Click **Add**.

   Add Sandbox server panel appears in the right part of the web interface window.

5. In the **IP** field, specify the IP address of the Sandbox server.
6. Click **Get certificate fingerprint**. Kaspersky Next XDR Expert displays the *Sandbox certificate fingerprint*. You will need this fingerprint later to match with the one displayed in the Sandbox web interface.
7. In the **Sandbox name** field, specify the Sandbox component name to be displayed in the OSMP interface.
8. Click **Save**.

OSMP sends a connection request to Sandbox. Now you need to accept the request.

To accept a connection request from OSMP:

1. In the Sandbox web interface, go to the **Authorization** section.
2. Compare the certificate fingerprints:

   - Sandbox certificate fingerprint displayed under Sandbox IP should match the Sandbox certificate fingerprint that you obtained earlier in the OSMP interface.
   - In the list of connection requests from OSMP, find the connection request from OSMP. The **Certificate fingerprint** column contains the certificate fingerprint of the OSMP server that has sent the connection request. This certificate fingerprint should match the OSMP certificate fingerprint you obtained earlier in the OSMP interface.

3. To accept the connection request, click **Accept**. Later you can revoke a previously accepted connection request by clicking the **Revoke** button.

4. Click **Apply**.

Integration with Sandbox is ready.

Kaspersky Next XDR Expert maintenance

This section describes updating, removing, and reinstalling Kaspersky Next XDR Expert components by using KDT. Also, the section provides instructions on how to stop the Kubernetes cluster nodes, update custom certificates for public Kaspersky Next XDR Expert services, as well as obtain the current version of the configuration file, and perform other actions with Kaspersky Next XDR Expert components by using KDT.

Kaspersky Next XDR Expert components

The table below lists the components included with Kaspersky Next XDR Expert.

Kaspersky Next XDR Expert components

Component name	Description
`agentserver`	Component that allows performing response actions on assets through agent integration.
`asset-manager`	Component for centralized asset management.
`backup-operator`	Automated tool that manages data backup and recovery via Custom Resources and labels.
`bootstrap`	Basic execution environment that includes the Kubernetes cluster and infrastructure components for the function of Kaspersky Next XDR Expert.
`cp`	Communication platform that enables data exchange and interaction between system components.
`csi-provider`	Component that enables cluster access to external storage via the Container Storage Interface (CSI).
`database-cluster`	Set of database servers that work together for high availability and scalability.
`database-operator`	CloudNativePG Kubernetes controller for managing internal database connection configuration, database resources, secrets, and credential rotation.
`gateway`	Kubernetes cluster gateway.
`hierman`	Component that manages hierarchy operations.
`iam`	OSMP platform component that provides authentication and authorization.
`irp`	Component that automates alert and incident handling, and executes response actions.
`istio`	Service mesh for managing service communication in Kubernetes clusters.
`keswin_web_plugin_xdr`	Web plug-in for managing Kaspersky Endpoint Security for Windows within Kaspersky XDR, supporting security policy configuration, incident monitoring, and platform component integration.
`ksb-facade`	Component that provides execution of actions by using mixins and deployment templates.
`ksc`	Kaspersky Security Center Administration Server.
`ksn-proxy-facade`	Component that manages KSN Proxy deployment via Porter, external KSN service integration, and statistics processing.
`kuma-migrator`	Component that automates migration of configurations, data, and settings between KUMA versions.
`loadbalancer`	Component that routes requests to containers or processes in the cluster, and provides auditing, authentication, and session synchronization across nodes.
`nsqlookupd`	Service that discovers NSQ producers and directs clients within the cluster.

Component name	Description
`nwc`	OSMP Console.
`objectstorage`	Component for storing temporary playbook data, alert and incident files, and supporting the S3 API for external integrations.
`observability`	Component that provides centralized visibility into events, threats, and assets, and supports response automation through integration with the KUMA platform.
`orchestra`	Automation platform. A common component for orchestrating workflows and executing playbooks to support automated incident response and streamline security operations.
`osmp-ksn-proxy`	Component that transfers detection and alert statistics to KSN or KPSN via the KSN proxy API and routes data.
`osmp-scanner`	Component that scans files by using Anti-Malware, YARA, and Sandbox, sends them for analysis, and generates reports.
`plugin-admin-agent`	Component that manages Network Agent policies.
`plugin-admin-server`	Component that manages Administration Servers.
`plugin-agent-server`	Component that manages the features of Kaspersky Endpoint Detection and Response.
`plugin-asset-management`	Component that manages assets, including devices, users, and other resources.
`plugin-enterprise`	Component that integrates with other Kaspersky solutions and provides asset management, monitoring, response automation, and multitenancy support via web interface and API.
`plugin-kata`	Component that integrates with KATA/EDR for response actions, incident data enrichment, and high availability and scalability support.
`plugin-kuma`	Component that integrates KUMA features into the Kaspersky XDR.
`plugin-osmp-scanner`	Component that provides scanning interaction, integrates with Anti-Malware, YARA, and Sandbox, sends results to ObjectStorage, Proxy, and Disaster Recovery, and supports automatic and manual response actions.
`plugin-public-api-token-settings`	Component that manages API token settings via the public API to support authentication and authorization for Kaspersky XDR components and external systems.
`plugin-tenant-roles`	Component that manages user and group role assignments within tenants and integrates with IAM for role model implementation and multitenancy support.
`spp_web_plugin_xdr`	Component that integrates with the Kaspersky Single Product Plugin to manage other Kaspersky applications through OSMP Console.
`secrets-manager`	Component that manages secrets, ensuring their secure storage and proper use during service deployment and configuration.
`tenant-registry`	Component that manages tenant lifecycle, identity data, and integration with Kaspersky Security Center servers and external systems.
`plugin-kessw`	Component for managing the Kaspersky Embedded Systems Security for Windows (KESS) within Kaspersky XDR.
`plugin-kes-linux`	Component that manages Kaspersky Endpoint Security for Linux.
`plugin-kesw`	Component that manages Kaspersky Endpoint Security for Windows.
`plugin-spp`	Kaspersky Single Product Plugin for managing other Kaspersky products through the OSMP Console.

Upgrading Kaspersky Next XDR Expert from version 1.4 to 2.0

You can upgrade Kaspersky Next XDR Expert from version 1.4 to 2.0 by using KDT. KDT updates the Kubernetes cluster, the Kaspersky Next XDR Expert components, and the KUMA services installed on the KUMA target hosts outside the cluster. When upgrading, you do not need to fill in the configuration file, because the installation parameters specified during deployment are used. The steps of the Kaspersky Next XDR Expert upgrade process do not depend on the selected deployment option.

To upgrade Kaspersky Next XDR Expert:

1. Download and unpack the distribution package with Kaspersky Next XDR Expert version 2.0.

2. On the administrator host, export the current configuration file.

3. In the exported configuration file, update the installation parameters listed in the table below.

Changed installation parameters

Old installation parameter (version 1.4)	New installation parameter (version 2.0)	Action
—	`kuma_roles` (only for multi-node deployment)	Add a new parameter
—	`grafana_smtp_host`	
—	`grafana_smtp_user`	
—	`grafana_smtp_password`	
—	`grafana_smtp_bundle`	
—	`grafana_smtp_skip_verify`	
—	`grafana_smtp_from_address`	
—	`grafana_smtp_from_name`	
—	`grafana_smtp_ehlo_identity`	
—	`grafana_starttls_policy`	
—	`grafana_smtp_enable_tracing`	
—	`xdr_edr_ingress_host`	
`vault_ha_mode` (only for single-node deployment)	`openbao_ha_mode` (only for single-node deployment)	Rename the parameters
`vault_standalone` (only for single-node deployment)	`openbao_standalone` (only for single-node deployment)	
`vault_ha_mode` (only for single-node deployment)	`openbao_ha_mode` (only for single-node deployment)	Rename the parameters
`vault_standalone` (only for single-node deployment)	`openbao_standalone` (only for single-node deployment)	

4. Read the End User License Agreement (EULA) of KDT located in the distribution package with the Kaspersky Next XDR Expert components.

   When you start using KDT, you accept the terms of the EULA of KDT.

   You can read the EULA of KDT after upgrading Kaspersky Next XDR Expert. The `license_kdt.txt` file is located in the `~/kaspersky_eula` directory of the user who runs the upgrade of Kaspersky Next XDR Expert.

5. Run the following command to upgrade Kaspersky Next XDR Expert and the Bootstrap component:

   ```
   ./kdt apply -k <full_path_to_XDR_updates_archive> --force
   ```

   To start upgrading Kaspersky Next XDR Expert, accept the terms of the EULA and the Privacy Policy.

   You can install Kaspersky Next XDR Expert without prompting to read the terms of the EULA and the Privacy Policy of OSMP, if you use the `--accept-eula` flag. In this case, you must read the EULA and the Privacy Policy of OSMP before upgrading Kaspersky Next XDR Expert. The files are located in the distribution package with the Kaspersky Next XDR Expert components. By using the `--accept-eula` flag, you confirm that you have fully read, understand, and accept the terms of the EULA and the Privacy Policy.

   If you want to read and accept the terms of the EULA and the Privacy Policy during the upgrade, do not use the `--accept-eula` flag.

6. If you do not use the `--accept-eula` flag in the previous step, read the EULA and the Privacy Policy of OSMP. The text is displayed in the command line window. Press the space bar to view the next text segment. Then, when prompted, enter the following values:

   a. Enter `y` if you understand and accept the terms of the EULA.

      Enter `n` if you do not accept the terms of the EULA.

   b. Enter `y` if you understand and accept the terms of the Privacy Policy, and if you agree that your data will be handled and transmitted (including to third countries) as described in the Privacy Policy.

      Enter `n` if you do not accept the terms of the Privacy Policy.

   If you do not accept the terms of the EULA and the Privacy Policy, Kaspersky Next XDR Expert will not be upgraded.

Kaspersky Next XDR Expert is upgraded from version 1.4 to 2.0.

Updating Kaspersky Next XDR Expert components

KDT allows you to update the Kaspersky Next XDR Expert components (including management web plug-ins) and the Bootstrap component. New versions of the Kaspersky Next XDR Expert components are included in the distribution package.

> Installing components of an earlier version is not supported.

To update the Kaspersky Next XDR Expert components:

1. Download the distribution package with the new versions of the Kaspersky Next XDR Expert components.

2. If necessary, on the administrator host, export the current version of the configuration file.

   You do not need to export the configuration file if the installation parameters are not added or modified.

3. Update the Kaspersky Next XDR Expert components and Bootstrap.

   Specify the path to the transport archive with the Kaspersky Next XDR Expert components and the path to the configuration file. The configuration file path is optional if it is specified in the Bootstrap update command or no installation parameters are added or modified:

   - Run the following command for standard updating of the Kaspersky Next XDR Expert components:

     ```
     ./kdt apply -k <full_path_to_XDR_updates_archive> -i
     <full_path_to_configuration_file>
     ```

   - If the version of the installed Kaspersky Next XDR Expert component matches the component version in the distribution package, the update of this component is skipped. Run the following command to force an update of this component by using the `force` flag:

     ```
     ./kdt apply --force -k <full_path_to_XDR_updates_archive> -i
     <full_path_to_configuration_file>
     ```

4. Read the End User License Agreement (EULA) and the Privacy Policy of the Kaspersky Next XDR Expert component, if a new version of the EULA and the Privacy Policy appears. The text is displayed in the command line window. Press the space bar to view the next text segment. Then, when prompted, enter the following values:

   a. Enter `y` if you understand and accept the terms of the EULA.

      Enter `n` if you do not accept the terms of the EULA. To use the Kaspersky Next XDR Expert component, you must accept the terms of the EULA.

   b. Enter `y` if you understand and accept the terms of the Privacy Policy, and you agree that your data will be handled and transmitted (including to third countries) as described in the Privacy Policy.

      Enter `n` if you do not accept the terms of the Privacy Policy.

> To update the Kaspersky Next XDR Expert component, you must accept the terms of the EULA and the Privacy Policy.

After you accept the EULA and the Privacy Policy, KDT updates the Kaspersky Next XDR Expert components.

You can read the EULA and the Privacy Policy of the Kaspersky Next XDR Expert component after the update. The files are located in the `~/kaspersky_eula` directory of the user who runs the deployment of Kaspersky Next XDR Expert.

Changing the Kaspersky Next XDR Expert parameters without reinstalling

If the organization's infrastructure has changed, then Kaspersky Next XDR Expert may need to change the main parameter values quickly, without reinstalling the solution. Kaspersky Next XDR Expert supports changing the Kubernetes cluster IP address (the `ingress_ip` parameter) and FQDN of services (in particular, changing the service domain name, the `smp_domain` parameter) by using KDT.

To change the `ingress_ip` parameter value without reinstalling Kaspersky Next XDR Expert,

On the administrator host, run the following command:

```
./kdt invoke gateway -a changeIngressIP --param ingress_ip='<new_ingress_ip>'
```

The `ingress_ip` parameter value is changed; and Kaspersky Next XDR Expert now uses the new parameter value.

To change the parameter values without reinstalling Kaspersky Next XDR Expert:

1. On the administrator host, export the current configuration file.

2. In the exported configuration file, change the `smp_domain` and `ingress_ip` parameter values.

3. Apply the parameter changes by using the command:

   ```
   ./kdt apply -i '<full_path_to_updated_configuration_file>'
   ```

   The `smp_domain` and `ingress_ip` parameter values are changed; and Kaspersky Next XDR Expert now uses the new parameter values.

When the value of the `smp_domain` parameter changes, the FQDN service names also change. Therefore, KDT reissues all certificates required for working with public Kaspersky Next XDR Expert services.

Adding and deleting primary and worker nodes of the Kubernetes cluster

If you need to scale the Kubernetes cluster to meet your current workload, increase its fault tolerance, or optimize computing resources, you can add or delete primary and worker nodes included in the cluster by using KDT.

> You can add or delete primary and worker nodes only if Kaspersky Next XDR Expert is deployed on multiple nodes.

Adding new primary and worker nodes to the Kubernetes cluster

To add new primary and worker nodes:

1. Prepare target hosts that will act as primary and worker nodes.

2. On the administrator host, export the current configuration file by using the following command:

   ```
   ./kdt ec -e '<configuration_file_name_with_path>'
   ```

   The current version of the configuration file is saved to the specified directory, with the specified name.

3. In the `nodes` section of the exported configuration file, add parameters of one or several new nodes (`desc`, `type`, `host`, `kind`, `user`, and `key`), and then save the configuration file.

4. Copy the public key to each new node (for example, to the `~/.ssh` directory) by using the ssh-copy-id utility.

5. On the administrator host, run the following command to apply the modified configuration file to the Kubernetes cluster. In the command, specify the full path to this configuration file:

   ```
   ./kdt apply -i '<full_path_to_configuration_file>'
   ```

   After you start the command, KDT checks whether the configuration file is filled in correctly. Then, KDT checks whether the hardware, software, and network configuration of the Kubernetes cluster nodes meet the prerequisites for installing the solution. If all the strict pre-checks are successfully completed, KDT outputs the notification of successful validation to stdout and KDT logs. Otherwise, logs with information about the failed validation are saved and the process of adding new hosts is interrupted. You can skip the pre-checks before you run the command, if needed (set the `ignore_precheck` installation parameter to `true`).

New primary and worker nodes are added to the Kubernetes cluster. KDT outputs the information about added primary and worker nodes to stdout and KDT logs.

Deleting primary and worker nodes from the Kubernetes cluster

When deleting nodes, take into account that the minimum allowed number of primary nodes in the cluster is one and the number of workers is three. If fewer than one primary and three workers remain when deleting, the operation will be interrupted. You can view the number of primary and worker nodes in the configuration file that will be obtained after export.

To delete primary and worker nodes:

1. On the administrator host, export the current configuration file by using the following command:

   ```
   ./kdt ec -e '<configuration_file_name_with_path>'
   ```

   The current version of the configuration file is saved to the specified directory, with the specified name.

2. In the `nodes` section of the exported configuration file, delete parameters of one or several primary and worker nodes (`desc`, `type`, `host`, `kind`, `user`, and `key`), and then save the configuration file.

   > Ensure that after deleting nodes, you will have at least one primary and three worker nodes left in the cluster. A minimum cluster configuration for the multi-node deployment includes four nodes (one primary node and three worker nodes). If this condition is not met, the deletion process is interrupted.

   > You can also delete a node with `kind=admsrv`, if it is not the only remaining node with `kind=admsrv` in the cluster. If you attempt to delete this node from the cluster, it will be skipped during the deletion process.

3. On the administrator host, run the following command to apply the modified configuration file to the Kubernetes cluster. In the command, specify the full path to this configuration file:

   ```
   ./kdt apply -i '< full_path_to_configuration_file >'
   ```

   The specified nodes are deleted. KDT outputs the information about deleted primary and worker nodes to stdout and KDT logs.

Versioning the configuration file

When working with Kaspersky Next XDR Expert, you may need to change the parameters that were specified in the configuration file before the Kaspersky Next XDR Expert deployment. For example, when changing the disk space used to store the Administration Server data, the `ksc_state_size` parameter is modified. The current version of the configuration file with the modified `ksc_state_size` parameter is updated in the Kubernetes cluster.

If you try to use the previous version of the configuration file in a KDT custom action that requires the configuration file, a conflict occurs. To avoid conflicts, you have to use only the current version on the configuration file exported from the Kubernetes cluster.

To export the current version of the configuration file,

On the administrator host where the KDT utility is located, run the following custom action, and then specify the path to the configuration file and its name:

```
./kdt ec -e <configuration_file_name_with_path>
```

The current version of the configuration file is saved to the specified directory with the specified name.

You can use the exported configuration file, for example, when updating Kaspersky Next XDR Expert components or adding management plug-ins for Kaspersky applications.

> You need not export the configuration file if the installation parameters are not added or modified.

Uninstalling Kaspersky Next XDR Expert

KDT allows you to uninstall all Kaspersky Next XDR Expert components installed in the Kubernetes cluster, the cluster itself, the KUMA services installed outside the cluster, and other artifacts created during the deployment or operation of the solution.

To uninstall the Kaspersky Next XDR Expert components and related data:

1. On the administrator host, run the following command:

   ```
   ./kdt remove --all
   ```

   This command removes the following objects and artifacts:

   - All Kaspersky Next XDR Expert components installed in the Kubernetes cluster and the cluster itself.
   - The Kaspersky Next XDR Expert user account that was created by KDT during the deployment.
   - The database located on a separate server or inside the cluster, DBMS schemes, and DBMS accounts created by KDT during the deployment.
   - The KUMA services installed outside the cluster on the hosts that were specified in the inventory file.
   - The contents of the following directories on target hosts:
     - `/var/spool/ksc/logs`
     - `/var/spool/ksc/backup`
     - `/var/spool/ksc/`
     - `/var/lib/k0s`
     - `/run/k0s`
     - `/etc/k0s/`
     - `/etc/containerd/`
     - `/var/lib/containerd/`
     - `/run/containerd/`
   - Logs obtained during the installation and operation of the Kaspersky Next XDR Expert components.
   - Data related to the Kaspersky Next XDR Expert components on the administrator host.

If the administrator host does not have network access to a target host, uninstalling the components is interrupted. You can restore network access and restart the uninstallation of Kaspersky Next XDR Expert. Alternatively, you can uninstall the Kaspersky Next XDR Expert components from the target hosts manually.

> If you use multiple Kubernetes clusters managing by contexts, this command removes only the current Kubernetes context, the corresponding cluster, and the Kaspersky Next XDR Expert components installed in the cluster. Other contexts and their clusters with Kaspersky Next XDR Expert instances are not removed.

2. Close the ports used by Kaspersky Next XDR Expert that were opened during the deployment, if needed. These ports are not closed automatically.

3. Remove the operating system packages that were automatically installed during the deployment, if needed. These packages are not removed automatically.

4. Remove KDT and the contents of the `~/kaspersky_eula` and `~/.kdt` directories.

The Kaspersky Next XDR Expert components, database, and related data are removed, and the ports used by Kaspersky Next XDR Expert are closed.

Kaspersky applications installed on the managed devices are not removed by using the `./kdt remove` command. For information on how to remove Kaspersky applications, refer to their documentation.

After uninstalling Kaspersky Next XDR Expert, target hosts are not restarted automatically. You can restart these hosts manually if necessary.

Manual uninstalling of Kaspersky Next XDR Expert components

If the administrator host does not have network access to a target host, uninstalling the Kaspersky Next XDR Expert components by using KDT is interrupted. You can restore network access and restart the removal of the solution. Alternatively, you can uninstall the Kaspersky Next XDR Expert components from the target hosts manually.

To uninstall the Kaspersky Next XDR Expert components from the target hosts manually:

1. On the target host, run the following command to stop the k0s service:

   ```
   /usr/local/bin/k0s stop
   ```

2. Run the following command to reset cluster nodes to the initial state:

   ```
   /usr/local/bin/k0s reset
   ```

3. Remove the contents of the following directories:

   - Required directories:

     - `/etc/k0s/`

     - `/var/lib/k0s/`

- `/usr/libexec/k0s/`
- `/usr/local/bin/` (remove only the `k0s` file)

- Optional directories:
  - `/var/lib/containerd/`
  - `/var/cache/k0s/`
  - `/var/cache/kubelet/`
  - `/var/cache/containerd/`

  You can remove the `/var/lib/containerd/` and `/var/cache/containerd/` directories if the containerd service is used only for the function of Kaspersky Next XDR Expert. Otherwise, your data contained in the `/var/lib/containerd/` and `/var/cache/containerd/` directories may be lost.

  The contents of the `/var/cache/k0s/`, `/var/cache/kubelet/`, and `/var/cache/containerd/` directories are automatically removed after you restart the target host. You do not have to clear these folders manually.

4. Restart all target hosts.

The Kaspersky Next XDR Expert components are deleted from the target hosts.

Reinstalling Kaspersky Next XDR Expert after a failed installation

If the installation of Kaspersky Next XDR Expert failed, you can restart the installation. Installation failure may occur during installation of Bootstrap or installation of Kaspersky Next XDR Expert components. The steps needed to reinstall Kaspersky Next XDR Expert depends on whether the Kubernetes cluster was deployed when the installation error occurred.

To determine the stage at which the installation failed,

Run the following command to view the list of all installed Kaspersky Next XDR Expert components:

```
./kdt state
```

If the Bootstrap installation error occurs, KDT displays a message that Bootstrap is not installed. In this case, you have to manually uninstall the directories created during the deployment.

If Bootstrap is installed successfully, it will have the `Success` status in the command output. This means that a Kaspersky Next XDR Expert components installation error occurred. In this case, you have to reinstall the components that were not installed correctly or reinstall the entire solution. In the command output, such components have the `Failed` status.

Reinstalling Kaspersky Next XDR Expert if Bootstrap is not installed

To restart the Kaspersky Next XDR Expert installation:

1. Go to the directory with the KDT utility, and view the installation logs of the Bootstrap component.

   If a failure occurred during the Bootstrap installation stage, this will be reflected in the log. In this case, Bootstrap has not been installed and the Kubernetes cluster has not been created.

2. Manually uninstall the directories created during the deployment.

   a. On the administrator host, go to the `.kdt` directory, and then remove the `cluster.config` file.

   b. Remove the contents of the following directories:

   - Required directories:
     - `/etc/k0s/`
     - `/var/lib/k0s/`
     - `/usr/libexec/k0s/`
     - `/usr/local/bin/` (remove only the `k0s` file)
   - Optional directories:
     - `/var/lib/containerd/`
     - `/var/cache/k0s/`
     - `/var/cache/kubelet/`
     - `/var/cache/containerd/`

     You can remove the `/var/lib/containerd/` and `/var/cache/containerd/` directories if the containerd service is used only for the function of Kaspersky Next XDR Expert. Otherwise, data contained in the `/var/lib/containerd/` and `/var/cache/containerd/` directories may be lost.

     The contents of the `/var/cache/k0s/`, `/var/cache/kubelet/`, and `/var/cache/containerd/` directories are automatically removed after you restart the target host. You do not have to clear these folders manually.

   c. Restart all target hosts.

3. Restart the installation of Kaspersky Next XDR Expert.

Reinstalling Kaspersky Next XDR Expert if Bootstrap is installed

To reinstall incorrectly installed Kaspersky Next XDR Expert components:

1. View the full installation log of the incorrectly installed Kaspersky Next XDR Expert component by using the command:

   ```
   ./kdt state -l <component_name>
   ```

You can also output all diagnostic information about Kaspersky Next XDR Expert components by using the command:

```
./kdt logs get --to-archive
```

2. Troubleshoot an installation error on your own or with the help of Kaspersky Technical Support.

3. Reinstall incorrectly installed Kaspersky Next XDR Expert components.

   - If you did not modify the configuration file, run the following command, and then specify the same transport archive that was used for the Kaspersky Next XDR Expert installation:

     ```
     ./kdt apply -k <full_path_to_transport_archive>
     ```

   - If you need to change the installation parameters, export the configuration file, modify it, and then run the following command with the transport archive and the updated configuration file:

     ```
     ./kdt apply -k <full_path_to_transport_archive> -i
     <full_path_to_configuration_file>
     ```

   KDT reinstalls only the incorrectly installed Kaspersky Next XDR Expert components.

4. Restart the installation of Kaspersky Next XDR Expert by using the command:

   ```
   ./kdt apply --force -k <full_path_to_XDR_updates_archive> -i
   <full_path_to_configuration_file>
   ```

If reinstalling the components did not fix an installation error, you can uninstall the entire solution.

To restart the Kaspersky Next XDR Expert installation:

1. Output all diagnostic information about Kaspersky Next XDR Expert components by using the following command:

   ```
   ./kdt logs get --to-archive
   ```

2. Troubleshoot an installation error on your own or with the help of Kaspersky Technical Support.

3. Uninstall Kaspersky Next XDR Expert. On the administrator host, run the command:

   ```
   ./kdt remove --all
   ```

4. Restart the installation of Kaspersky Next XDR Expert by using the command:

   ```
   ./kdt apply --force -k <full_path_to_XDR_updates_archive> -i
   <full_path_to_configuration_file>
   ```

Reinstalling Kaspersky Next XDR Expert components

During the installation of Kaspersky Next XDR Expert, on the administrator host, KDT displays an installation log that shows whether the Kaspersky Next XDR Expert components are installed correctly.

After installing Kaspersky Next XDR Expert, you can run the following command to view the list of all installed components:

```
./kdt state
```

The installed components list is displayed. Correctly installed components have the `Success` status. If the component installation failed, this component has the `Failed` status.

To view the full installation log of the incorrectly installed Kaspersky Next XDR Expert component, run the following command:

```
./kdt state -l <component_name>
```

You can also output all diagnostic information about Kaspersky Next XDR Expert components by using the following command:

```
./kdt logs get --to-archive
```

You can use the obtained logs to troubleshoot problems on your own or with the help of Kaspersky Technical Support.

To reinstall incorrectly installed Kaspersky Next XDR Expert components,

- If you did not modify the configuration file, run the following command, and then specify the same transport archive that was used for the Kaspersky Next XDR Expert installation:

  ```
  ./kdt apply -k <full_path_to_transport_archive>
  ```

- If you need to change the installation parameters, export the configuration file, modify it, and then run the following command with the transport archive and the updated configuration file:

  ```
  ./kdt apply -k <full_path_to_transport_archive> -i <full_path_to_configuration_file>
  ```

KDT reinstalls only the incorrectly installed Kaspersky Next XDR Expert components.

Operation diagnostics of the Kaspersky Next XDR Expert components

This section describes how to obtain diagnostic information about Kaspersky Next XDR Expert components.

Obtaining log files of Kaspersky Next XDR Expert components

KDT allows you to obtain log files that contain diagnostic information about Kaspersky Next XDR Expert components and the Kubernetes cluster, to troubleshoot problems on your own or with the help of Kaspersky Technical Support.

Kaspersky Next XDR Expert generates the log file names according to the following template: pod_name.container_name.log. Here, pod_name is a Kubernetes pod name, and container_name is a Kubernetes container name.

To obtain log files of Kaspersky Next XDR Expert components and management web plug-ins,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt logs get <flags>
```

Where `<flags>` are the parameters of the command that allow you to configure the logging result.

You can specify the following logging parameters:

- `--app <list_of_components>`—Obtain logs for the listed Kaspersky Next XDR Expert components.

  For example, if you want to obtain logs for the last hour for Administration Server, run the following command:

  ```
  ./kdt logs get --app ksc
  ```

- `--auto-dest-dir`—Obtain logs and save them to the kdt-default-logs-<current_date_and_time> directory that is automatically created in the current directory. If the logging period is not specified, you obtain diagnostic information for the last hour.

  For example, if you want to obtain logs for the last hour for Administration Server and KUMA, and then save these logs to the automatically created directory, run the following command:

  ```
  ./kdt logs get --app ksc,core --auto-dest-dir
  ```

- `-d, --destination <file_path>`—Obtain logs and save them to the specified file.

- `-D, --destination-dir <directory_path>`—Obtain logs and save them to the specified directory that must be created beforehand. If the `<directory_path>` is empty, logs are saved in the standard output stream (stdout). If the logging period is not specified, you obtain diagnostic information for the last hour.

- `--to-archive`—Obtain logs and save them to the kdt-default-logs-<current_date_and_time>.tar.gz archive. The created archive is saved to the current directory. If the logging period is not specified, you obtain diagnostic information for the last hour.

- `--last=<hours>h`—Obtain logs for the specified number of hours up to date.

  For example, if you want to get an archive with logs for the last three hours, run the following command:

  ```
  ./kdt logs get --to-archive --last=3h
  ```

- `--start=<date_and_time>`—Obtain logs starting from the specified date and time (in the Unix timestamp format) to the present time, or to the date and time specified in the `--end` parameter.

  For example, if you want to obtain logs starting from 03/26/2024 10:00:00 to the present time, and then save them to the kdt-default-logs-<current_date_and_time> directory created in the current directory, run the following command:

  ```
  ./kdt logs get --auto-dest-dir --start=1711447200
  ```

- `--end=<date_and_time>`—Obtain logs starting from the date and time specified in the `--start` parameter to the date and time specified in the `--end` parameter (in the Unix timestamp format). If the `--start` parameter in not specified, logs are obtained for the last hour before the date and time specified by the `--end` parameter.

  For example, if you want to save logs for the 10 minutes (from 03/26/2024 10:00:00 to 03/26/2024 10:10:00) to the logs directory, run the following command:

  ```
  ./kdt logs get -D ./logs/ start=1711447200 --end=1711447800
  ```

To view the available logging parameters, you can run one of the following commands:

- `./kdt logs get -h`

- `./kdt logs get --help`

Viewing OSMP metrics

The observability component of Kaspersky Next XDR Expert allows you to view OSMP metrics, monitor the state of each component by dashboards, view metrics of the data storage system (Longhorn) for the Kubernetes cluster, and notify users by email when the Kubernetes cluster node performance indicators exceed normal values.

You can monitor metrics for further analysis of the operability and performance of its components by using the observability component.

You can view OSMP metrics in one of the following ways:

- By using Grafana dashboards
- By using your own monitoring tools

To view OSMP metrics by using Grafana dashboards:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

   The `<monitoring_host>` is a host name, and `<smp_domain>` is a domain name. These parameters constitute the FQDN of the Kaspersky Next XDR Expert monitoring service and are set in the configuration file when deploying Kaspersky Next XDR Expert.

2. Enter the Grafana credentials specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

The Grafana interface is displayed. You can view diagnostic information for each Kaspersky Next XDR Expert component, view metrics of the data storage system (Longhorn) for the Kubernetes cluster, and configure email notifications about Kubernetes cluster health events.

To view OSMP metrics by using your own monitoring tools:

1. Configure your monitoring tools to obtain the metrics from the Kaspersky Next XDR Expert API.

2. Go to the API address `<api_host>.<smp_domain>/api/read-api` to obtain metrics.

   The `<api_host>` is a host name, and `<smp_domain>` is a domain name. These parameters constitute the FQDN of the service that provide access to the Kaspersky Next XDR Expert API. These parameters are set in the configuration file when deploying Kaspersky Next XDR Expert.

OSMP metrics in the OpenMetrics format are obtained.

If you want to view information about the performance of the KUMA Core, storage, collectors, and correlators, you have to view KUMA metrics.

Monitoring the state of Kaspersky Next XDR Expert components

The dashboard provides a graphical display of the state of each Kaspersky Next XDR Expert component.

For example, you can view the following component parameters:

- Usage of requests and limits of CPU
- Usage of requests of CPU and RAM
- Usage of CPU and RAM by containers
- Allocation of the component resources by containers
- Network performance indicators:
  - Bandwidth
  - Packets rate (number of received packets)
  - Packets dropped (number of lost packets)
  - Errors (number of network errors)

To view diagnostic information on the dashboard:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

   The `<monitoring_host>` is a host name, and `<smp_domain>` is a domain name. These parameters constitute the FQDN of the Kaspersky Next XDR Expert monitoring service and are set in the configuration file when deploying Kaspersky Next XDR Expert. For example: `monitoring.smp.local`.

2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

3. In the menu, go to **Home** → **Dashboards** → **Kubernetes / Views / Pods**.

4. In the **namespace** drop-down list, select the component for which you want to view the diagnostic information.

5. You can also specify other parameters to customize the dashboard view.

   The dashboard with diagnostic information about the selected Kaspersky Next XDR Expert component is displayed.

Configuring email notifications about Kubernetes cluster health events

The observability component of the Kaspersky Next XDR Expert allows you to notify users by email when the Kubernetes cluster node performance indicators exceed normal values. Email notifications are generated based on email notification rules in Grafana. These rules define the notification message, recipients, and conditions under which email alerts are sent.

You can view the list of available email notification rules in Grafana. Rules can be either predefined or custom. Predefined rules are provided by Kaspersky Next XDR Expert and cannot be changed. Custom rules can be created, modified, or deleted.

Predefined rules for generating email notifications

Kaspersky Next XDR Expert includes the following predefined rules for generating email notifications about Kubernetes cluster health events. You cannot delete or edit these rules.

Predefined email notifications

Metric	Severity	Notification message
High CPU Usage	Warning	CPU usage on node <node_name> has exceeded the warning threshold. This may indicate elevated load or insufficient CPU resources.
	Critical	Node <node_name> is experiencing critically high CPU usage. This can lead to CPU throttling, pod-scheduling issues, and degraded performance of cluster components.
High Disk I/O Utilization	Warning	Disk device <disk_name> on node <node_name> is under elevated I/O load. Applications may start seeing latency.
	Critical	Disk device <disk_name> on node <node_name> is overloaded. This can severely degrade performance of stateful applications.
High Disk Latency	Warning	Read/write latency on storage device <disk_name> on node <node_name> has crossed the warning threshold.
	Critical	Storage device <disk_name> on node <node_name> is experiencing critically high I/O latency.
High Disk Usage (Filesystem)	Warning	Filesystem <mountpoint_name> on node <node_name> has reached a high utilization level. Approaching the limit may affect logging, pod scheduling, or kubelet stability.
	Critical	Filesystem <mountpoint_name> on node <node_name> is almost full. This can cause write errors, service failures, and node instability.
High Memory Usage	Warning	Memory usage on node <node_name> has exceeded the warning threshold. Workloads may encounter memory pressure.
	Critical	Node <node_name> is near memory exhaustion. This may trigger OOMKill or pod eviction. Immediate action is required to prevent workload failures.

To view predefined rules for generating email notifications:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

   The `<monitoring_host>` is a host name, and `<smp_domain>` is a domain name. These parameters constitute the FQDN of the Kaspersky Next XDR Expert monitoring service and are set in the configuration file when deploying Kaspersky Next XDR Expert. For example: `monitoring.smp.local`.

2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

3. In the menu, go to **Home** → **Alerting** → **Alert rules**.

In the **Alert rules** section, the list of predefined rules for generating email notifications is displayed.

Custom rules for generating email notifications

You can create your own rules for generating email notifications about Kubernetes cluster health events.

Email notifications will not be sent if SMTP server connection parameters are not specified in the configuration file: `grafana_smtp_host`, `grafana_smtp_user`, `grafana_smtp_password`, `grafana_smtp_bundle`, `grafana_smtp_skip_verify`, `grafana_smtp_from_address`, `grafana_smtp_from_name`, `grafana_smtp_ehlo_identity`, `grafana_starttls_policy`, `grafana_smtp_enable_tracing`. If you did not specify these parameters in the configuration file before deploying Kaspersky Next XDR Expert, you can add them later.

To create custom rules for generating email notifications:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

   The `<monitoring_host>` is a host name, and `<smp_domain>` is a domain name. These parameters constitute the FQDN of the Kaspersky Next XDR Expert monitoring service and are set in the configuration file when deploying Kaspersky Next XDR Expert. For example: `monitoring.smp.local`.

2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

3. In the menu, go to **Home** → **Alerting** → **Alert rules**.

4. In the **Alert rules** section, click **New alert rule**, and then specify the parameters of a new email notification rule.

5. After you specify the rule parameters, click **Save rule and exit**.

The custom rule is created. You can also create folders and place the created rules in them.

To create custom rules for generating email notifications from a dashboard:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

3. In the menu, go to **Home** → **Dashboards**.

4. In the dashboard list, click the name of the dashboard that contains performance indicators you want to use for alerting.

   The dashboard opens.

5. Click the **Menu** icon (⋮) on the required panel of the dashboard, and then in the context menu, select **More...** → **New alert rule**.

6. In the **New alert rule** panel, specify the parameters of a new email notification rule.

7. Click **Save rule and exit**.

The custom rule from the dashboard is created.

To edit or delete custom rules for generating email notifications:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

3. In the menu, go to **Home** → **Alerting** → **Alert rules**.

4. In the **Alert rules** section, locate the required rule, and then do one of the following:

   - To edit the rule, click the pencil button, and then change the rule parameters.
   - To delete the rule, click **More**, and then click **Delete**.

   The custom rule is changed or deleted.

Managing email notification recipients for Kubernetes cluster health events

Kaspersky Next XDR Expert allows you to specify, edit, and delete notification recipient email addresses.

> Email notifications will not be sent if SMTP server connection parameters are not specified in the configuration file: `grafana_smtp_host`, `grafana_smtp_user`, `grafana_smtp_password`, `grafana_smtp_bundle`, `grafana_smtp_skip_verify`, `grafana_smtp_from_address`, `grafana_smtp_from_name`, `grafana_smtp_ehlo_identity`, `grafana_starttls_policy`, `grafana_smtp_enable_tracing`. If you did not specify these parameters in the configuration file before deploying Kaspersky Next XDR Expert, you can add them later.

To add a notification recipient email address:

1. Go to the `<monitoring_host>.<smp_domain>` URL.

   The `<monitoring_host>` is a host name, and `<smp_domain>` is a domain name. These parameters constitute the FQDN of the Kaspersky Next XDR Expert monitoring service and are set in the configuration file when deploying Kaspersky Next XDR Expert. For example: `monitoring.smp.local`.

2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).

3. In the menu, go to **Home** → **Alerting** → **Contact points**.

4. In the **Contact points** workspace, click **Add contact point**, and then specify the following recipient parameters:

   - **Name**—Name of the recipient or group of recipients. Required field.
   - **Integration**—Type of integration. Use Email for email notifications.
   - **Addresses**—One or multiple recipient email addresses. Addresses can be separated by a comma, new line, or semicolon (`,` or `\n` or `;`).
   - **Single email**—Option to send a single email to all recipients. Possible parameter values:
     - **Enabled**—All recipients receive one single email notification.
     - **Disabled**—Each recipient receives a separate email notification.
   - **Message**—Email body. You can use templates or set custom text by clicking **Edit Message**.
   - **Subject**—Email subject. You can use templates or set custom text by clicking **Edit Subject**.
   - **Notification settings**—General notification settings.

- **Disable resolved message**—Disables sending a message when an issue or alert is resolved.
- **Test**—Sending a test notification. Allows checking that settings are correct.
- **Duplicate**—Creating a copy of the contact point configuration. Allows quickly creating similar contact points.
- **Delete**—Deleting the current contact point configuration.

5. Click **Save contact point**.

To edit or delete a notification recipient email address:

1. Go to the `<monitoring_host>.<smp_domain>` URL.
2. Enter the Grafana credentials that you specified in the configuration file (the `grafana_admin_user` and `grafana_admin_password` parameters).
3. In the menu, go to **Home** → **Alerting** → **Contact points**.
4. In the **Contact points** workspace, locate the required contact, and then do one of the following:
   - To edit the rule, click the pencil button, and then change the rule parameters.
   - To delete the rule, click **More**, and then click **Delete**.

The contact is changed or deleted.

Adding SMTP server connection parameters to the configuration file without reinstalling Kaspersky Next XDR Expert

To add SMTP server connection parameters to the configuration file without reinstalling Kaspersky Next XDR Expert:

1. On the administrator host, export the current configuration file.
2. In the exported configuration file, add and specify the following parameters in the `parameters` section: `grafana_smtp_host`, `grafana_smtp_user`, `grafana_smtp_password`, `grafana_smtp_bundle`, `grafana_smtp_skip_verify`, `grafana_smtp_from_address`, `grafana_smtp_from_name`, `grafana_smtp_ehlo_identity`, `grafana_starttls_policy`, `grafana_smtp_enable_tracing`.
3. Apply the parameter changes by using the command:

   ```
   ./kdt apply -i '<full_path_to_updated_configuration_file>'
   ```

   SMTP server connection parameters are added to the configuration file.

Viewing Longhorn metrics

Kaspersky Next XDR Expert includes a dashboard that displays key metrics for the cluster's data storage system (Longhorn). The Longhorn dashboard enables monitoring of data storage health and performance.

By default, the dashboard displays data for the last three hours, but you can change this period. The dashboard refreshes every 15 seconds.

To view the Longhorn dashboard:

1. Go to `<monitoring_host>.<smp_domain>`.

   `<monitoring_host>` represents a host name, and `<smp_domain>` represents a domain name. Together, they form the FQDN of the Kaspersky Next XDR Expert monitoring service and are specified in the configuration file during Kaspersky Next XDR Expert deployment.

2. Enter the Grafana credentials that you specified in the configuration file (`grafana_admin_user` and `grafana_admin_password`).

3. In the menu, go to **Home** → **Dashboardrs** → **Longhorn Dashboard**.

   The Longhorn dashboard is displayed.

The Longhorn dashboard metrics are described below.

Nodes

- **Nodes**—total number of nodes in the system.
- **Schedulable Nodes**—number of nodes available for distributing volume replicas.
- **Disabled Nodes**—nodes disabled by a user.
- **Failed Nodes**—unresponsive, offline or otherwise unavailable node.

Volumes

- **Attached Volumes**—volumes currently attached to nodes.
- **Detached Volumes**—volumes not attached to nodes.
- **Degraded Volumes**—volumes with an insufficient number of replicas.
- **Faulty Volumes**—volumes without functioning replicas.

Capacity

- **Node Capacity**—total storage capacity on each node.
- **Disk Capacity**—storage capacity on each disk.
- **Volume Capacity**—capacity of each volume.

Resource usage

- **Node Storage Usage/Capacity**—percentage of storage capacity on each node.
- **Disk Space Usage**—percentage of capacity on each disk.
- **Volumes Per Node**—number of mounted volumes on each node.
- **Volume Size/Capacity**—current/maximum volume ratio.

- **Volume Write Throughput**—amount of data the system writes per second (bps).
- **Volume Read Throughput**—amount of data the system reads per second (bps).
- **Volume Write IOPS**—number of write input/output operations (IOPS).
- **Volume Read IOPS**—number of read input/output operations (IOPS).
- **Volume Write Latency**—time required to complete a single write operation (ns).
- **Volume Read Latency**—time required to complete a single read operation (ns).
- **Node CPU Usage/Capacity**—percentage of CPU utilization per node.
- **Node Memory Usage/Capacity**—percentage of memory utilization per node.
- **Longhorn Manager CPU Usage**—CPU load of the Longhorn Manager Longhorn (milliCPU).
- **Longhorn Manager Memory Usage**—memory load of the Longhorn Manager Longhorn.
- **Instance Manager CPU Usage**—CPU load of Longhorn instances (milliCPU).
- **Instance Manager Memory Usage**—memory used by Longhorn instances.

Disk usage

The indicators on the disk usage panel display the percentage of time the disk spends reading and writing. These indicators are based on the following metrics:

- `longhorn_volume_read_iops`
- `longhorn_volume_read_latency_seconds`
- `longhorn_volume_write_iops`
- `longhorn_volume_write_latency_seconds`

Viewing EDR dashboards and metrics

Kaspersky Next XDR Expert features EDR dashboards that display key metrics for components responsible for processing objects from Endpoint Agents. Monitoring these metrics helps ensure the components operate smoothly and securely, enabling proactive measures to prevent potential issues.

The following dashboards are available:

- EDR Agent Server
- EDR KSB
- EDR KSN proxy

- EDR Sandbox
- EDR Scanner

To view the EDR dashboards:

1. Go to `<monitoring_host>.<smp_domain>`.

   `<monitoring_host>` represents a host name, and `<smp_domain>` represents a domain name. Together, they form the FQDN of the Kaspersky Next XDR Expert monitoring service and are set in the configuration file when deploying Kaspersky Next XDR Expert.

2. Enter the Grafana credentials you specified in the configuration file (`grafana_admin_user` and `grafana_admin_password`).

3. In the menu, go to **Home** → **Dashboards**.

4. In the tag filtering drop-down menu (**Filter by tag**), select the **edr** tag.

   The list of available EDR dashboards is displayed.

5. Click the name of a dashboard you want to view.

   The selected dashboard opens.

Widgets and metrics available in each EDR dashboard are described below.

EDR Agent Server

By default, the dashboard displays data for the last two days. You can change this period by clicking **Last 2 days** at the top of the page and selecting a different time range.

EDR Agent Server widgets and metrics

Widget name	Metric name	Description
Tasks	xdr_agentserver_tasks_total	Total number of tasks received from Incident Response Platform.
Settings	xdr_agentserver_settings_total	Total number of configurations received from Incident Response Platform.
Active tasks	xdr_agentserver_tasks_active	Total number of tasks in progress.
Active settings	xdr_agentserver_settings_active	Total number of configuration applications in progress.
Tasks errors	xdr_agentserver_tasks_errors_total	Total number of tasks completed with an error.
Settings errors	xdr_agentserver_settings_errors_total	Total number of configuration application errors.
Object storage	xdr_agentserver_objectstorage_upload_errors_total	Total number of file upload errors to Object Storage.
Object storage	xdr_agentserver_objectstorage_download_errors_total	Total number of file download errors from Object Storage.
Agent info update	xdr_agentserver_agentinfo_updates_total	Total number of times Endpoint Agent information is updated and sent to the Asset Manager.
Assets resolve errors	xdr_agentserver_assets_resolve_errors_total	Total number of errors when mapping identifiers between OSMP and Endpoint Agents.
Assets	xdr_agentserver_assets_resolve_total	Total number of resolutions.
Agent server rate limit exceeded	xdr_agentserver_rate_limit_exceeded_total	Total number of cases when rate limit was exceeded.
Agent auth	xdr_agentserver_agent_auth_total	Total number of Endpoint Agent authentication attempts.
Agent auth	xdr_agentserver_agent_auth_errors_total	Total number of Endpoint Agent authentication errors.

Widget name	Metric name	Description
Vtproxy sessions	xdr_agentserver_vtproxy_sessions_total	Total number of sessions created.
Vtproxy auth errors	xdr_agentserver_vtproxy_auth_errors_total	Total number of session authentication errors.
Vtproxy successful auth	xdr_agentserver_vtproxy_successful_auths_total	Total number of successful session authentications.
Vtproxy active sensor sessions	xdr_agentserver_vtproxy_sensor_session_active	Active sessions for sensor.
Vtproxy active sessions	xdr_agentserver_vtproxy_sessions_active	Number of active sessions.
Vtproxy sessions duration	xdr_agentserver_vtproxy_session_duration_seconds	Session duration in seconds.
Vtproxy bytes transferred	xdr_agentserver_vtproxy_bytes_transferred_total	Total number of bytes transferred.

EDR KSB

By default, the dashboard displays data for the last two days. You can change this period by clicking **Last 2 days** at the top of the page and selecting a different time range.

EDR KSB widgets and metrics

Widget name	Metric name	Description
Multitasks	xdr_ksb_multitasks_total	Total number of scan tasks.
Active multitasks	xdr_ksb_multitasks_active	Number of active scan tasks.
Multitasks duplicates	xdr_ksb_multitask_duplicates_total	Number of duplicate scan tasks.
Multitasks files duplicates	xdr_ksb_multitask_file_duplicates_total	Number of duplicate files in scan tasks.
Multitasks files size	xdr_ksb_multitask_files_size_bytes_total	Size of multitask files in bytes.
Agents auth failures	xdr_ksb_agent_auth_failures_total	Number of Endpoint Agent authentication failures.
Rate limit exceeded	xdr_ksb_rate_limit_exceeded_total	Total number of cases when rate limit was exceeded.

EDR KSN proxy

By default, the dashboard displays data for the last six hours. You can change this period by clicking **Last 6 hours** at the top of the page and selecting a different time range.

EDR KSN proxy widgets and metrics

Widget name	Metric name	Description
Stats messages	xdr_scanner_ksnproxyfacade_stats_messages_total	Total number of statistics messages.
Integration mode changes	xdr_scanner_ksnproxyfacade_integration_mode_changes_total	Total number of integration type changes.
KSN Proxy starts	xdr_scanner_ksnproxysupervisor_ksnproxy_starts_total	Total number of KSN proxy starts.
KSN Proxy start errors	xdr_scanner_ksnproxysupervisor_ksnproxy_start_errors_total	Total number of KSN proxy start errors.
KSN availability	xdr_scanner_ksnproxy_ksn_proxy_ksn_available	KSN availability.
KSN heartbeat failed requests	xdr_scanner_ksnproxy_ksn_proxy_ksn_heartbeat_failed_requests_total	Total number of failed heartbeat requests.
KSN heartbeat requests	xdr_scanner_ksnproxy_ksn_proxy_ksn_heartbeat_requests_total	Total number of heartbeat requests.

EDR Sandbox

By default, the dashboard displays data for the last six hours. You can change this period by clicking **Last 6 hours** at the top of the page and selecting a different time range.

To view the metrics of a certain Sandbox, select its IP address in the **Sandbox** menu at the top.

EDR Sandbox widgets and metrics

Widget name	Metric name	Description
Start time Uptime	xdr_scanner_sb_process_start_time_seconds	Time when the Sandbox process started and its uptime, in seconds.
SB last health check SB health check errors	xdr_scanner_sb_health_check_errors_current	Time of last health check and current number of detected errors.
SB bases age	xdr_scanner_sb_health_bases_age_seconds	Age of anti-virus databases in Sandbox, in seconds.
CPU	xdr_scanner_sb_process_cpu_seconds_total	Number of CPU cores used.
Memory usage	xdr_scanner_sb_go_memstats_sys_bytes	Memory requested from the system, in bytes.
Memory usage	xdr_scanner_sb_go_memstats_heap_inuse_bytes	Heap memory in use, in bytes.
Memory usage	xdr_scanner_sb_go_memstats_stack_inuse_bytes	Stack memory in use, in bytes.
Memory usage	xdr_scanner_sb_process_resident_memory_bytes	Resident memory (RSS), in bytes.
Disk writes/reads	xdr_scanner_sb_node_disk_read_bytes_total	Total bytes read from disk.
Disk writes/reads	xdr_scanner_sb_node_disk_written_bytes_total	Total bytes written to disk.
Network usage	xdr_scanner_sb_process_network_receive_bytes_total	Incoming traffic.
Network usage	xdr_scanner_sb_process_network_transmit_bytes_total	Outgoing traffic.
Malware channel availability	xdr_scanner_sbscanapi_sb_malware_channel_availability	Availability of malware channel for specific Sandbox.
SB images status	xdr_scanner_sb_vm_images_status	Current status of virtual machines in the Sandbox, indicating whether operating systems are installed.
SB image version status	xdr_scanner_sb_vm_image_version_status	The status of image versions used on virtual machines, indicating whether each version is correct.

EDR Scanner

By default, the dashboard displays data for the last two days. You can change this period by clicking **Last 2 days** at the top of the page and selecting a different time range.

In the **Scan engine** menu at the top, select the scanning engine to view its metrics:

- **AM**: Anti-Malware Engine metrics
- **YARA**: YARA metrics
- **SB**: Sandbox metrics

EDR Scanner widgets and metrics

Widget name	Metric name	Description
Last bases update	xdr_scanner_updater_updater_bases_age	Updater bases age.
Last bases update	xdr_scanner_updater_updater_retranslation_age	Updater retranslation age.
Last bases update	xdr_scanner_internalscanner_am_bases_age	Age of anti-virus databases for Anti-Malware Engine.

Widget name	Metric name	Description
Last bases update	xdr_scanner_sb_health_bases_age_seconds	Time elapsed since the last update of the anti-virus databases in Sandbox, in seconds.
AM scan engine complete SB scan engine complete YARA scan engine complete	xdr_scanner_internalscanner_scan_engine_complete_total	Number of completed scan tasks for each scanning engine and object scanning results.
Scan health	xdr_scanner_internalscanner_total_scans	Number of scans.
Scan health	xdr_scanner_internalscanner_failed_scans	Number of failed scans.
AM scan tasks	xdr_scanner_internalscanner_am_queue_len	Number of jobs in the Anti-Malware Engine queue.
AM scan tasks	xdr_scanner_internalscanner_am_queue_total_items	Total number of tasks in all Anti-Malware Engine queues.
AM scan tasks	xdr_scanner_internalscanner_am_queue_failed_items_total	Total number of failed tasks in Anti-Malware Engine queues.
AM queue size	xdr_scanner_internalscanner_am_queue_size	File system size used by the Anti-Malware Engine queue.
AM queue size	xdr_scanner_internalscanner_am_storage_size_limit	Maximum allowed size of Anti-Malware Engine queue.
SB scan tasks	xdr_scanner_internalscanner_sb_wait_queue_len	Number of jobs in the Sandbox queue in OSMP scanner.
SB scan tasks	xdr_scanner_internalscanner_sb_queue_total_items	Total number of tasks in all Sandbox queues.
SB scan tasks	xdr_scanner_internalscanner_sb_queue_failed_items_total	Total number of failed tasks in Sandbox queues.
SB scan tasks	xdr_scanner_internalscanner_sb_queue_len	Number of tasks in Sandbox queue.
SB queue size	xdr_scanner_internalscanner_sb_queue_size	Approximate size of data (samples) in Sandbox queue.
SB queue size	xdr_scanner_internalscanner_sb_storage_size_limit	Maximum allowed size of Sandbox queue.
Average sandbox scan time	xdr_scanner_sbscanapi_sb_scan_time_total	Total Sandbox scanning time.
Average sandbox scan time	xdr_scanner_sbscanapi_sb_scan_count_total	Total number of scans on Sandbox.
YARA scan tasks	xdr_scanner_internalscanner_yr_queue_len	Number of jobs in the YARA queue.
YARA queue size	xdr_scanner_internalscanner_yr_queue_size	File system size used by YARA queue.
YARA queue size	xdr_scanner_internalscanner_yr_storage_size_limit	Maximum YARA queue size.
YARA scan tasks	xdr_scanner_internalscanner_yr_queue_total_items	Total number of tasks in all YARA queues.
YARA scan tasks	xdr_scanner_internalscanner_yr_queue_failed_items_total	Total number of failed tasks in YARA queues.
Last bases update	xdr_scanner_internalscanner_am_bases_age	Age of anti-virus databases for Anti-Malware Engine.

Viewing the Node Exporter Full dashboard

The Node Exporter Full dashboard displays key metrics for cluster node performance.

To view the Node Exporter Full dashboard:

1. Go to `<monitoring_host>.<smp_domain>`.

   `<monitoring_host>` represents the host name, and `<smp_domain>` represents the domain name. Together, they form the FQDN of the Kaspersky Next XDR Expert monitoring service and are specified in the configuration file when deploying Kaspersky Next XDR Expert.

2. Enter the Grafana credentials that you specified in the configuration file (`grafana_admin_user` and `grafana_admin_password`).

3. In the menu, go to **Home** → **Dashboards**.

4. Click **Node Exporter Full**.

   The dashboard opens.

By default, the dashboard displays data for the last 24 hours. You can change this period by clicking **Last 24 hours** at the top of the page and selecting a different period.

Panes available in the dashboard are described in the table below.

Node Exporter Full panes

Pane	Description
Quick CPU / Mem / Disk	Quick overview of key resources: CPU load, memory usage, and disk usage.
Pressure	Pressure Stall Information (PSI): CPU, memory, and I/O. Shows where the system is under load.
CPU Busy	Percentage of time processors are busy with computations (user + system time).
Sys Load	Average system load (load average) over 1, 5, and 15 minutes.
RAM Used	Used RAM, including buffers and cache.
SWAP Used	Used swap space.
Root FS Used	Space used on the root file system (/).
CPU Cores	Number of cores (vCPU) on the node.
Uptime	Time elapsed since the last system boot.
RootFS Total	Total size of the root file system.
RAM Total	Total amount of RAM.
SWAP Total	Total amount of available swap memory.
Basic CPU / Mem / Net / Disk	Overview of key CPU, memory, network, and disk metrics.
CPU Basic	Basic CPU load panel, including user/system/idle/iowait metrics.
Memory Basic	Main RAM and swap usage metrics.
Network Traffic Basic	Total network traffic (inbound/outbound) by interface.
Disk Space Used Basic	Disk space usage and partitioning.
CPU	Detailed CPU usage metrics (in percent, by core).
Memory Stack	Diagram of memory layer usage (used, cache, buffers, free).
Network Traffic	Total network traffic (in bytes/second) by interface.
Disk Space Used	Detailed disk space usage by partition.
Disk IOps	Number of input/output operations per second (I/O per second).
I/O Usage Read / Write	Data read and write speed on disks (bytes/second).
I/O Utilization	Storage device utilization: percentage of time the disk is busy with I/O.
CPU spent seconds in guests (VMs)	CPU time spent on running guest virtual machines (guest time).
Memory Active / Inactive	Active and inactive memory pages (used and potentially reclaimable).
Memory Committed	Amount of memory allocated to processes (including overcommit).
Memory Active / Inactive Detail	Details of active and inactive pages, including anon/file.
Memory Writeback and Dirty	Amount of pages waiting to be written to disk (dirty, writeback).
Memory Shared and Mapped	Shared and mapped memory.

Pane	Description
Memory Slab	Memory allocated to kernel objects (slab, reclaimable/unreclaimable).
Memory Vmalloc	Kernel virtual memory allocation.
Memory Bounce	Number of bounce pages (rare, for old hardware).
Memory Anonymous	Anonymous memory (not file-backed).
Memory Kernel / CPU	Memory used by kernel and CPU structures.
Memory HugePages Counter	Number of huge pages used (HugePages).
Memory HugePages Size	Size and state of HugePages.
Memory DirectMap	Direct mapping of physical memory (DirectMap).
Memory Unevictable and MLocked	Memory locked in RAM (mlocked, unevictable).
Memory NFS	Memory usage by the NFS subsystem.
Memory Meminfo	Summary information from /proc/meminfo.
Memory Pages In / Out	Number of pages loaded/unloaded into memory.
Memory Pages Swap In / Out	Pages moved between RAM and swap.
Memory Page Faults	Number of page faults (minor/major faults).
OOM Killer	Events where system killed processes due to memory shortage.
Memory Vmstat	Extended virtual memory statistics (/proc/vmstat).
Time Synchronized Drift	Difference between system time and synchronization source.
Time PLL Adjust	Time adjustment parameters (PLL, offset).
Time Synchronized Status	Time synchronization status (for example, chrony/ntpd).
Time Misc	Other time-related metrics (offset, jitter).
System Timesync	Overall time synchronization status on the node.
Processes Status	Process statistics by state (running, sleeping, zombie).
Processes State	Histogram of process states.
Processes Forks	Number of processes created (forks).
Processes Memory	Memory usage by processes.
PIDs Number and Limit	Current and maximum number of PIDs in the system.
Process schedule stats Running / Waiting	Process distribution by "running" and "waiting" states.
Threads Number and Limit	Number of threads and system limit.
Context Switches / Interrupts	Context switches and hardware interrupts.
System Load	Average load (load average) over 1, 5, and 15 minutes.
CPU Frequency Scaling	Current CPU frequency and scaling (governor, min/max).
Pressure Stall Information	CPU, memory, and I/O pressure - shows task delays.
Interrupts Detail	Detailed breakdown of interrupt types.
Schedule timeslices executed by each cpu	Number of time quanta executed by each core.
Entropy	Available entropy (random number generation quality).
CPU time spent in user and system contexts	Total CPU time spent in user and system modes.
File Descriptors	Number of open file descriptors.
Hardware temperature monitor	Temperature of system components (CPU, disks, sensors).
Throttle cooling device	State of cooling systems and throttling.
Power supply	Power data (voltage, current, battery).
Systemd Sockets	Active systemd sockets.

Pane	Description
Systemd Units State	State of systemd units (active, failed, inactive).
Disk IOps Completed	Completed I/O operations.
Disk R/W Data	Amount of data read/written.
Disk Average Wait Time	Average I/O wait time.
Average Queue Size	Average I/O queue length.
Disk R/W Merged	Number of merged read/write operations.
Time Spent Doing I/Os	Time spent on I/O operations.
Instantaneous Queue Size	Current I/O queue length.
Disk IOps Discards completed / merged	Discard (TRIM) operations completed/merged.
Filesystem space available	Available space on file systems.
File Nodes Free	Number of free inodes.
File Descriptor	Number of file descriptors.
File Nodes Size	Inode table size.
Filesystem in ReadOnly / Error	File system state (errors, readonly mount).
Network Traffic by Packets	Number of Rx/Tx packets by interface.
Network Traffic Errors	Transmission and reception errors.
Network Traffic Drop	Dropped packets.
Network Traffic Compressed	Compressed traffic.
Network Traffic Multicast	Multicast traffic.
Network Traffic Fifo	FIFO errors on interfaces.
Network Traffic Frame	Frame errors.
Network Traffic Carrier	Link-level errors (carrier).
Network Traffic Colls	Network collisions.
NF Conntrack	Number of tracked network connections (conntrack).
ARP Entries	ARP table (number of entries).
MTU	Maximum packet size (MTU).
Speed	Network interface speed (bits/second).
Queue Length	interface packet queue length.
Softnet Packets	Softnet packet processing (kernel).
Softnet Out of Quota	Dropped packets due to softnet overflow.
Network Operational Status	Current interface state (up/down).
Sockstat TCP / UDP / FRAG / RAW	Statistics for TCP, UDP, and other protocol sockets.
Sockstat Memory Size / Used	Memory usage by sockets and number of active sockets.
Netstat IP In / Out Octets	Number of incoming/outgoing IP packets (in bytes).
Netstat IP Forwarding	IP forwarding status.
ICMP In / Out / Errors	ICMP packet and error statistics.
UDP In / Out / Errors	UDP packet and error statistics.
TCP In / Out / Errors	TCP traffic and errors.
TCP Connections	Number of active TCP connections by state.
TCP SynCookie / Direct Transition / Stat	SYN cookie protection, direct TCP state transitions, and total TCP statistics.
Node Exporter Scrape Time / Scrape	Node Exporter performance, with focus on scrape durations and target status.

Service monitoring

You can use the following functionality to monitor the status of all services except cold storage and agent:

- Viewing Victoria Metrics alerts

  Users with the General administrator role can configure thresholds for KUMA services, and if a specified threshold is exceeded, the following changes take place:

  - KUMA logs an audit event: VictoriaMetrics alert registered for service.
  - KUMA sends a notification email message to the General administrator.
  - Services are displayed in the **Active services** section with a yellow status. If you hover over the status icon, the error message will be displayed.

    Possible service statuses

    - Green means the service is running and accessible from the Core server.
    - Red means the service is not running or is not accessible from the Core server.
    - Yellow is the status that applies to all services except the agent. The yellow status means that the service is running, but there are errors in the service log, or there are alerts for the service from Victoria Metrics. You can view the error message by hovering the mouse cursor over the status of the service in the **Active services** section.
    - Purple is the status that is applied to running services whose configuration file in the database has changed, but that have no other errors. If a service has an incorrect configuration file and has errors, for example, from Victoria Metrics, status of the service is yellow.
    - Gray means that if a deleted tenant had a running service that is still running, that service is displayed with a gray status on the Active services page. Services with the gray status are kept when you delete the tenant to let you copy the ID and remove services on your servers. Only the General administrator can delete services with the gray status. When a tenant is deleted, the services of that tenant are assigned to the Main tenant.

- View VictoriaMetrics metrics if the user has a role with metrics access rights.

The following examples show how to monitor service status.

1. If the collector service has a yellow status in the **Active services** section and you see the **Enrichment errors increasing** message, you can:

   - Go to **Monitoring** → **Metrics** → <service type> → <service name> → **Enrichment** → **Errors** section for the service with the yellow status, find out which enrichment is causing errors, and view the chart to find out when the problem started and how it evolved.
   - The likely cause of the enrichment errors may be DNS server unavailability or CyberTrace enrichment errors, therefore you can check your DNS or CyberTrace connection settings.

2. If the collector service has a yellow status in the **Active services** section and you see the **Output Event Loss increasing** message, you can:

   - Go to the **Monitoring** → **Metrics** → <service type> → <service name> → **IO** → **Output Event Loss** section for the service with the yellow status and view the chart to find out when the problem started and how it evolved.

   - The likely cause of the enrichment errors may be a buffer overflow or unavailability of destination, therefore you can check the availability and the connection of the destination or find out why the buffer capacity is exceeded.

Configuring service monitoring

To configure the services:

1. In the main application window, go to the **Settings** → **KUMA** → **Open KUMA settings** section.

   This opens the KUMA settings window.

2. Under Other, click **Services monitoring**.

   This opens the **Services monitoring** window.

3. Specify the values of monitoring parameters for the services.

   Service monitoring does not apply to cold storage.

   If you specify an invalid value that does not fit the range or format, the value is reset to the previously configured value.

4. Click **Save**.

   After saving the parameters, KUMA registers an audit event: Monitoring thresholds changed for the service.

KUMA monitors the status of services in accordance with the specified parameters.

In the **Active services** section, you can filter services by statuses or enter a word from the error text, for example, "QPS" or "buffer", in the search field and press ENTER. This results in a list of services with errors. Special characters ", },{, are not allowed in the search string and will produce irrelevant results.

Disabling service monitoring

To disable service monitoring:

1. In the main application window, go to the **Settings** → **KUMA** → **Open KUMA settings** section.

   This opens the KUMA settings window.

2. Under Other, click **Services monitoring**.

   This opens the **Services monitoring** window.

3. If you want to disable service monitoring only for collectors, in the **Service monitoring. Thresholds setting** window, under **Collectors**, select the **Disable connector errors** check box.

   This disables only the analysis of the **Connector errors** metric for collectors.

4. If you want to disable monitoring for all services, in the **Service monitoring. Thresholds setting** select the **Disable** check box.

KUMA service monitoring is disabled, and services do not get the yellow status.

Viewing KUMA metrics

To monitor the performance of its components, the event stream, and the correlation context, KUMA collects and stores a large number of parameters. The VictoriaMetrics time series database is used to collect, store and analyze the parameters. The collected metrics are visualized using Grafana. Dashboards that visualize key performance parameters of various KUMA components can be found in the **Monitoring** → **Metrics** section.
The KUMA Core service configures VictoriaMetrics and Grafana automatically, no user action is required.

The collected metrics are visualized using the Grafana solution. The RPM package of the 'kuma-core' service generates the Grafana configuration and creates a separate dashboard for visualizing the metrics of each service. Graphs in the **Metrics** section appear with a delay of approximately 1.5 minutes.

For full information about the metrics, you can refer to the **Metrics** section of OSMP Console. Selecting this section opens the Grafana portal that is deployed as part of Core installation and is updated automatically. If the **Metrics** section shows core: <port number>, this means that KUMA is deployed in a high availability configuration and the metrics were received from the host on which the **Core was installed**. In other configurations, the name of the host from which KUMA receives metrics is displayed.

To determine on which host the Core is running, run the following command in the terminal of one of the controllers:

```
k0s kubectl get pod -n kuma -o wide
```

Collector metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer Size	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.
Write Network BPS	The number of bytes received into the network per second.
Connector errors	The number of errors in the connector logs.

Normalization—metrics related to the normalizers.	
Raw & Normalized event size	The size of the raw event and size of the normalized event. The median value is displayed.
Errors	The number of normalization errors per second.

Filtration—metrics related to filters.	
EPS	The number of events per second matching the filter conditions and sent for processing. The collector only processes events that match the filtering criteria if the user has added the filter to the configuration of the collector service.

Aggregation—metrics related to the aggregation rules.	
EPS	The number of events received and generated by the aggregation rule per second. This metric helps determine the effectiveness of aggregation rules.
Buckets	The number of buckets in the aggregation rule.

Enrichment—metrics related to enrichment rules.	
Cache RPS	The number of requests per second to the local cache.
Source RPS	The number of requests per second to an enrichment source, such as a dictionary.
Source Latency	Time in milliseconds passed while sending a request to the enrichment source and receiving a response from it. The median value is displayed.
Queue	The size of the enrichment request queue. This metric helps to find bottleneck enrichment rules.
Errors	The number of errors per second while sending requests to the enrichment source.

Correlator metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer Size	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.

Correlation—metrics related to correlation rules.	
EPS	The number of correlation events per second generated by the correlation rule.
Buckets	The number of buckers in a correlation rule of the standard type.
Rate Limiter Hits	The number of times the correlation rule exceeded the rate limit per second.
Active Lists OPS	The number of operations requests per second sent to the active list, and the operations themselves.
Active Lists Records	The number of records in the active list.
Active Lists On-Disk Size	The size of the active list on the disk, in bytes.

Enrichment—metrics related to enrichment rules.	
Cache RPS	The number of requests per second to the local cache.
Source RPS	The number of requests per second to an enrichment source, such as a dictionary.

Source Latency	Time in milliseconds passed while sending a request to the enrichment source and receiving a response from it. The median value is displayed.
Queue	The size of the enrichment request queue. This metric helps to find bottleneck enrichment rules.
Errors	The number of errors per second while sending requests to the enrichment source.

Response—metrics associated with response rules.	
RPS	The number of times a response rule was activated per second.

Storage metrics

Metric name	Description
ClickHouse / General—metrics related to the general settings of the ClickHouse cluster.	
Active Queries	The number of active queries sent to the ClickHouse cluster. This metric is displayed for each ClickHouse instance.
QPS	The number of queries per second sent to the ClickHouse cluster.
Failed QPS	The number of failed queries per second sent to the ClickHouse cluster.
Allocated memory	The amount of RAM, in gigabytes, allocated to the ClickHouse process.

ClickHouse / Insert—metrics related to inserting events into a ClickHouse instance.	
Insert EPS	The number of events per second inserted into the ClickHouse instance.
Insert QPS	The number of ClickHouse instance insert queries per second sent to the ClickHouse cluster. If the **Insert QPS** metric demonstrates a growing queue of queries and the metric is greater than 1, we recommend also looking at the **Batch size** metric value to adjust the storage buffering settings in the settings of the storage service configuration. Example: The **Insert QPS** metric is greater than 1 and equals 8. The **Batch size** metric is 1.2 GB (in bytes). In this case, you can find the buffer size by multiplying **Insert QPS** by **Batch size**: 8 * 1.2 = 9.6 GB. Round the resulting value of 9.6 and specify it in bytes (for example, 10000000000 bytes) as the **Buffer size** setting on the **Advanced settings** tab of the storage service configuration. Also specify a **Buffer flush interval** of 2 seconds. Increasing the buffer size and buffer flush interval will help relieve the query queue. Normally, the **Insert QPS** metric should be less than 1.
Failed Insert QPS	The number of failed ClickHouse instance insert queries per second sent to the ClickHouse cluster.
Delayed Insert QPS	The number of delayed ClickHouse instance insert queries per second sent to the ClickHouse cluster. Queries were delayed by the ClickHouse node due to exceeding the soft limit on active merges.
Rejected Insert QPS	The number of rejected ClickHouse instance insert queries per second sent to the ClickHouse cluster. Queries were rejected by the ClickHouse node due to exceeding the hard limit on active merges.
Active Merges	The number of active merges.
Distribution Queue	The number of temporary files with events that could not be inserted into the ClickHouse instance because it was unavailable. These events cannot be found using search.

ClickHouse / Select—metrics related to event selections in the ClickHouse instance.	
Select QPS	The number of ClickHouse instance event select queries per second sent to the ClickHouse cluster.
Failed Select QPS	The number of failed ClickHouse instance event select queries per second sent to the ClickHouse cluster.

ClickHouse / Replication—metrics related to replicas of ClickHouse nodes.	
Active Zookeeper Connections	The number of active connections to the Zookeeper cluster nodes. In normal operation, this number should be equal to the number of nodes in the Zookeeper cluster.
Read-only Replicas	The number of read-only replicas of ClickHouse nodes. In normal operation, no such replicas of ClickHouse nodes must exist.
Active Replication Fetches	The number of active processes of downloading data from the ClickHouse node during data replication.
Active Replication Sends	The number of active processes of sending data to the ClickHouse node during data replication.
Active Replication Consistency Checks	The number of active data consistency checks on replicas of ClickHouse nodes during data replication.

ClickHouse / Networking—metrics related to the network of the ClickHouse cluster.	
Active HTTP Connections	The number of active connections to the HTTP server of the ClickHouse cluster.
Active TCP Connections	The number of active connections to the TCP server of the ClickHouse cluster.
Active Interserver Connections	The number of active service connections between ClickHouse nodes.

Core metrics

Metric name	Description
Raft—metrics related to reading and updating the state of the Core.	
Lookup RPS	The number of lookup procedure requests per second sent to the Core, and the procedures themselves.
Lookup Latency	Time in milliseconds spent running the lookup procedures, and the procedures themselves. The time is displayed for the 99th percentile of lookup procedures. One percent of lookup procedures may take longer to run.
Propose RPS	The number of Raft (SQLite) propose procedure requests per second sent to the Core, and the procedures themselves.
Propose Latency	Time in milliseconds spent running the Raft (SQLite) propose procedures, and the procedures themselves. The time is displayed for the 99th percentile of propose procedures. One percent of propose procedures may take longer to run.
Data mining—metrics related to data collection and analysis	
Executing Rules	Number of running schedulers for executing data collection and analysis requests.
Queued Rules	Number of schedulers for executing queued data collection and analysis requests.
Execution Errors	Number of errors that occurred when running the data collection and analysis scheduler.
Execution Latency	How long takes a scheduler to execute requests.
Tasks—metrics related to monitoring the running of tasks on the KUMA Core	
Active tasks	Number of tasks run per unit time.
Task Execution latency	Duration of running tasks in seconds.
Errors	Number of errors when running tasks.

API—metrics related to API requests.	
RPS	The number of API requests made to the Core per second.
Latency	The time in milliseconds spent processing a single API request to the Core. The median value is displayed.
Errors	The number of errors per second while sending API requests to the Core.

Notification Feed—metrics related to user activity.	
Subscriptions	The number of clients connected to the Core via SSE to receive server messages in real time. This number is normally equal to the number of clients that are using the KUMA web interface.
Errors	The number of errors per second while sending notifications to users.

Schedulers—metrics related to Core tasks.	
Active	The number of repeating active system tasks. The tasks created by the user are ignored.
Latency	The time in milliseconds spent running the task. The median value is displayed.
Errors	The number of errors that occurred per second while performing tasks.

KUMA agent metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer Size	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.
Write Network BPS	The number of bytes received into the network per second.

Event routers metrics

Metric name	Description
IO—metrics related to the service input and output.	
Processing EPS	The number of events processed per second.
Output EPS	The number of events per second sent to the destination.
Output Latency	The time in milliseconds that passed while sending an event packet and receiving a response from the destination. The median value is displayed.
Output Errors	The number of errors occurring per second while event packets were sent to the destination. Network errors and errors writing to the disk buffer of the destination are displayed separately.
Output Event Loss	The number of events lost per second. Events can be lost due to network errors or errors writing the disk buffer of the destination. Events are also lost if the destination responds with an error code, for example, in case of an invalid request.
Output Disk Buffer Size	The size of the disk buffer of the collector associated with the destination, in bytes. If a zero value is displayed, no event packets have been placed in the collector's disk buffer and the service is operating correctly.
Write Network BPS	The number of bytes received into the network per second.
Connector Errors	The number of errors in the connector log.

General metrics common for all services

Metric name	Description
Process—General process metrics.	
Memory	RAM usage (RSS) in megabytes.
DISK BPS	The number of bytes read from or written to the disk per second.
Network BPS	The number of bytes received/transmitted over the network per second.
Network Packet Loss	The number of network packets lost per second.
GC Latency	The time, in milliseconds, spent executing a GO garbage collection cycle The median value is displayed.
Goroutines	The number of active goroutines. This number is different from the operating system's thread count.

OS—metrics related to the operating system.	
Load	Average load.
CPU	CPU load as a percentage.
Memory	RAM usage (RSS) as a percentage.
Disk	Disk space usage as a percentage.

Metrics storage period

KUMA operation data is saved for 3 months by default. This storage period can be changed.

To change the storage period for KUMA metrics:

1. Log in to the OS of the server where the KUMA Core is installed.

2. In the file /etc/systemd/system/multi-user.target.wants/kuma-victoria-metrics.service, in the ExecStart parameter, edit the `--retentionPeriod=<metrics storage period, in months>` flag by inserting the necessary period. For example, `--retentionPeriod=4` means that the metrics will be stored for 4 months.

3. Restart KUMA by running the following commands in sequence:

   a. `systemctl daemon-reload`

   b. `systemctl restart kuma-victoria-metrics`

   The storage period for metrics has been changed.

KUMA metric alert triggering conditions

If the value of a KUMA metric for a service exceeds the threshold of the corresponding parameter configured in the **Services monitoring** section of KUMA, VictoriaMetrics sends an alert, and an error message is displayed in the status of that service.

Alerts are received from VictoriaMetrics at the following intervals:

- VictoriaMetrics collects information from KUMA services every 15 seconds.
- VictoriaMetrics updates alerts for KUMA services every minute.
- The KUMA Core service collects information from VictoriaMetrics every 15 seconds.

Thus, the total delay before a service status is updated is less than 2–3 minutes.

If you disabled the receipt of alerts from VictoriaMetrics, some KUMA services may still be displayed with a yellow status. This can happen in the following cases:

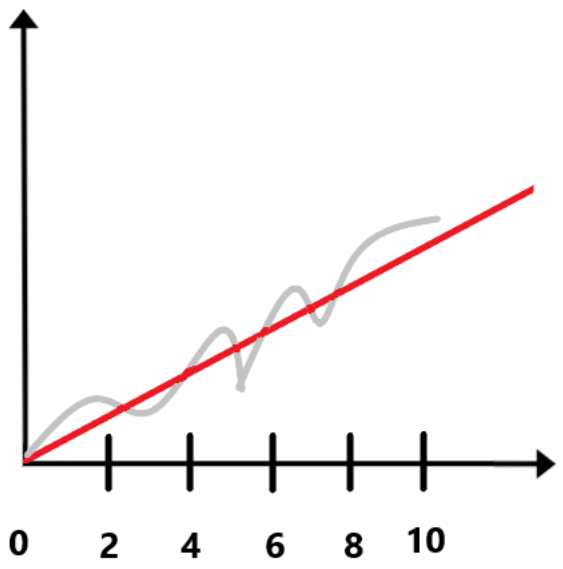
- For a storage service:
  - If an alert was received in response to an API request in the /status parameter from ClickHouse
  - If cold storage of the Storage service is not being monitored
- For a collector service: If an alert was received in response to an API request in the /status parameter.
- For a correlator service: If a response rule exists that requires the Advanced Responses module, but this module is not covered by the current license, or the license that covers this module has expired.

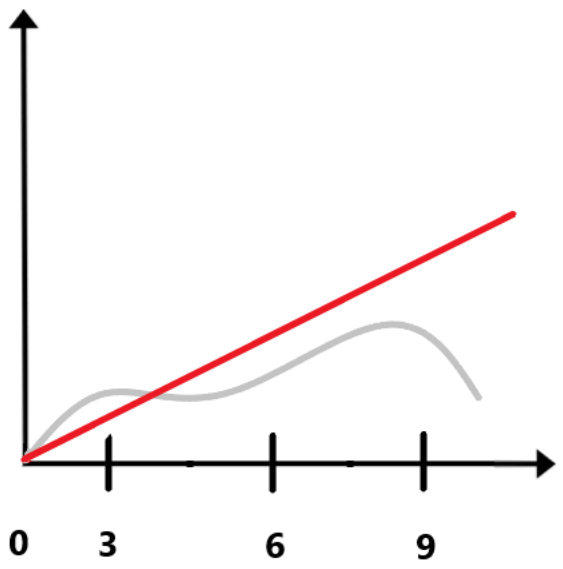
The table below provides information on which error messages may appear in the service status when an alert is received from VictoriaMetrics, and which metrics and parameters they are based on and in what way. For details on KUMA metrics that can trigger VictoriaMetrics alerts, see Viewing KUMA metrics.

For example, if the **Active services** table for a service displays a yellow status and the `High distribution queue` error message (the "Error message" column in the table below), you can view the information in the **Enrichment** widget, the **Distribution Queue** metric (the "KUMA metrics" column in the table below).

Description of error messages for KUMA services

Error message	Configurable alert parameters	KUMA metric	Description
`QPS threshold reached`	**QPS interval/window, minutes** **QPS Threshold**	Clickhouse / General → Failed QPS	An error message is displayed if the Failed QPS metric exceeds the specified **QPS Threshold** value for the duration specified by the **QPS interval/window, minutes** parameter. For example, if 25 out of 100 requests from VictoriaMetrics to the service were unsuccessful, and the **QPS Threshold** is 0.2, the alert is calculated as follows: (25 / 100) * 100 > 0.2 * 100 25% > 20% Because the percentage of unsuccessful requests is greater than the specified threshold, an error message is displayed for the service.
`Failed Insert QPS threshold reached`	**Failed insert QPS calculation interval/window, minutes** **Insert QPS threshold**	Clickhouse / Insert → Failed Insert QPS	An error message is displayed if the Failed Insert QPS metric exceeds the specified **QPS Insert Threshold** value for the duration specified by the **Failed Insert QPS calculation interval/window, minutes** parameter. For example, if 25 out of 100 requests from VictoriaMetrics to the service were unsuccessful, and the **QPS Insert Threshold** is 0.2, the alert is calculated as follows: (25 / 100) * 100 > 0.2 * 100 25% > 20% Because the percentage of unsuccessful requests is greater than the specified threshold, an error message is displayed for the service.

Error message	Configurable alert parameters	KUMA metric	Description
`High distribution queue`	**Distribution queue threshold** **Distribution queue calculation interval/window, minutes**	Clickhouse / Insert → Distribution Queue	An error message is displayed if the Distribution Queue metric exceeds the specified **Distribution queue threshold** value for the duration specified by the **Distribution queue calculation interval/window, minutes** parameter.
`Low disk space`	**Free space on disk threshold**	OS → Disk	An error message is displayed if the amount of free disk space (as a percentage) indicated by the Disk metric value is less than the value specified in the **Free disk space threshold** parameter. For example, an error message is displayed if the partition on which KUMA is installed takes up all the disk space.
`Low disk partition space`	**Free space on partition threshold**	OS → Disk	An error message is displayed if the amount of free space (as a percentage) on the disk partition that KUMA is using is less than the value specified in the **Free space on partition threshold** parameter. For example, an error message is displayed in the following cases: • If KUMA is installed in a high availability configuration, when the disk is mounted as a volume. • If the disk is mounted under /opt.
`Output Event Loss increasing`	**Output Event Loss**	IO → Output Event Loss	An error message is displayed if the Output Event Loss metric has been increasing for one minute. You can enable or disable the display of this error message using the **Output Event Loss** parameter.
`Disk buffer size increasing`	**Disk buffer increase interval/window, minutes**	IO → Output Disk Buffer SIze	An error message is displayed if the Output Disk Buffer Size metric monotonically increases for 10 minutes with the sampling interval specified by the **Disk buffer increase interval/window, minutes** parameter. For example, if the **Disk buffer increase interval/window, minutes** is set to 2 minutes, an error message is displayed if the disk buffer size has monotonically increased for 10 minutes with a sampling interval of 2 minutes (see the figure below). 

Error message	Configurable alert parameters	KUMA metric	Description
`High enrichment queue`	**Growing enrichment queue interval/window, minutes**	Enrichment → Queue	An error message is displayed if the Queue metric monotonically increases for 10 minutes with the sampling interval specified by the **Growing enrichment queue interval/window, minutes** parameter. For example, if the value of the **Growing enrichment queue interval/window, minutes** is 3, an error message is displayed if the enrichment queue has monotonically increased every 10 minutes with a sampling interval of 3 minutes. In the case shown in the figure below, the error message is not displayed because at the ninth minute the value of the metric decreased instead of increasing monotonically. 
`Enrichment errors increasing`	**Enrichment errors**	Enrichment → Errors	An error message is displayed if the Errors metric has been increasing for one minute. You can enable or disable the display of this error message using the **Enrichment errors** parameter.
`Connector log errors increasing`	**Disable connector errors**	IO → Connector Errors	An error message is displayed if the Connector Errors metric has been increasing between consecutive polls of the metric by VictoriaMetrics for one minute. You can enable or disable the display of this error message using the **Disable connector errors** parameter.

Storing diagnostic information about Kaspersky Next XDR Expert components

Diagnostic information about Kaspersky Next XDR Expert components is stored on a worker node of the Kubernetes cluster. The amount of disk space required for storing this information is specified in the configuration file before the deployment of Kaspersky Next XDR Expert (the `loki_size` parameter).

To check the disk space used to store diagnostic information about Kaspersky Next XDR Expert components,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke observability --action getPvSize
```

The amount of the allocated free disk space in gigabytes is displayed.

You can also increase the disk space used to store diagnostic information about Kaspersky Next XDR Expert components after the deployment of Kaspersky Next XDR Expert. You cannot set the amount of disk space to less than the previously specified amount.

To increase the disk space used to store diagnostic information about Kaspersky Next XDR Expert components,

On the administrator host where the KDT utility is located, run the following command and specify the required free disk space in gigabytes (for example, "50Gi"):

```
./kdt invoke observability --action setPvSize --param
loki_size='<new_disk_space_amount>Gi'
```

The amount of free disk space allocated to store diagnostic information about Kaspersky Next XDR Expert components is changed.

Obtaining trace files

KDT allows you to obtain trace files for Kaspersky Next XDR Expert and OSMP components, to troubleshoot infrastructure on your own or with the help of Kaspersky Technical Support.

Trace files are downloaded in OpenTelemetry format.

To obtain the trace file for the Kaspersky Next XDR Expert or OSMP component:

1. On the administrator host where the KDT utility is located, run the following command and specify the path to the file where you want to save the list of trace files:

   ```
   ./kdt traces find -o <output_file_path>
   ```

   The list of trace files with their IDs is output to the specified file.

2. To output a particular trace file run the following command and specify the output file path and the trace file ID:

   ```
   ./kdt traces get -o <output_file_path> --trace-id=<trace_ID>
   ```

The specified trace file is saved.

Logging the launches of custom actions

KDT allows you to obtain the history of the custom action launches for a specific Kaspersky Next XDR Expert component, as well as the logs of a particular custom action launch. The obtained logs may help you to investigate problems with the operation of the Kaspersky Next XDR Expert components on your own or with the help of Kaspersky Technical Support.

To obtain the history of the custom action launches for a specific Kaspersky Next XDR Expert component,

On the administrator host where the KDT utility is located, run the following command, and then specify the component name:

```
./kdt state -H <component_name>
```

The list of executed custom actions with their IDs is displayed.

To obtain logs of the custom action launch,

On the administrator host where the KDT utility is located, run the following command, and then specify the component name and the ID of the custom action launch:

```
./kdt state -l <component_name> -m <custom_action_launch_ID>
```

The logs of the specified custom action launch are displayed.

Backup copying and restoration of Kaspersky Next XDR Expert data

Data backup allows you to save a full copy of Kaspersky Next XDR Expert data. During the backup process, the operation of Kaspersky Next XDR Expert components are temporarily suspended to ensure data consistency. After the backup is created, it is saved in the network folder on a separate server, and then the solution resumes its functioning. You can create a backup manually or according to the specified schedule.

Following this scenario, you can configure the backup storage, set up the scheduled backup or create a backup manually, manage your backups, and restore Kaspersky Next XDR Expert data from the backup.

Limitations:

- During the backup process, the operation of Kaspersky Next XDR Expert components are temporarily suspended to ensure data consistency.
- The backup includes all Kaspersky Next XDR Expert data. Creating a backup or restoring data of an individual component are not supported.
- Restoring Kaspersky Next XDR Expert data from the backup must be performed by using the same version of Kaspersky Next XDR Expert in which the backup was created.
- Backing up and restoring Kaspersky Next XDR Expert data is supported only if the DBMS is not included in the Kubernetes cluster. Configuration where the DBMS is installed into the Kubernetes cluster is used only to get acquainted with the solution, not to solve standard work tasks.
- Free space for a backup is not checked automatically. You have to make sure that you have enough free space in the network folder.

Stages

The backup copying and restoration of Kaspersky Next XDR Expert data proceeds in stages:

1. **Configure the backup storage**

   Specify the path to the network folder for storing backups. This network folder must be located on a separate server and must be available from the Kubernetes cluster by using the CIFS or NFS protocol.

   To prevent an overload of the network folder, the maximum number of backups that can be stored in the network folder is limited. By default, you can create up to 10 backups. You can set the maximum number of backups by using the `backup_retain` parameter when creating a backup manually or when configuring a backup schedule. If the network folder already contains the maximum number of backups, the next backup will overwrite the oldest one.

Because the available space in the network folder isn't checked automatically, we recommend regularly monitoring the free space in this folder to prevent backup failures.

2. **Create a backup copy of Kaspersky Next XDR Expert data**

   Use one of the following ways to create a backup:

   - Scheduled backup

     If you want to create backups according to a schedule, you have to configure the schedule by using KDT. You can specify only one backup schedule. Creating a new schedule will replace the existing one. Additionally, you can view the current schedule, and pause or resume automatic backups.

     Before using the automatic scheduled backup, we recommend creating a backup manually to ensure that the network folder for storing backups is configured correctly and is accessible.

   - Manual backup

     If you need to save a backup without waiting for the next automatic backup, you can create a manual backup.

   A backup is saved to the specified network folder. You can view the list of available backups and details of each backup.

   KDT logs provide detailed information about the results of the backup operation, including its success or failure, the date and time of execution, and the name and location of the backup copy.

3. **Restore Kaspersky Next XDR Expert data from the backup copy**

   Restore the backup copy to roll back Kaspersky Next XDR Expert to a saved state.

Configuring the storage for a Kaspersky Next XDR Expert backup copy

All created backups are stored in the network folder on a separate server. This folder must be available from the Kubernetes cluster by using the CIFS or NFS protocol.

> Free space for a backup is not checked automatically. You have to make sure that you have enough free space on the network resource.

To prevent an overload of the network folder, the maximum number of backups that can be stored in the network folder is limited. By default, you can create up to 10 backups. You can set the maximum number of backups by using the `backup_retain` parameter when creating a backup manually or when configuring a backup schedule. If the network folder already contains the maximum number of backups, the next backup will overwrite the oldest one.

Specifying the path to the network folder for backup storage

To specify the backup storage path accessible via the CIFS/SMB protocol,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a setBackupStore \
  --param backup_target_url="cifs://<host>/<path>" \
  --param cifs_username="<username>" \
  --param cifs_password="<password>"
```

where:

- < `host` > is the address of the network storage server.
- < `path` > is the path to the backup network folder.
- < `username` > is the user name for accessing the network folder.
- < `password` > is the password for accessing the network folder.

  If the password contains double quotes, escape them with backslashes (`\"`).

To specify the backup storage path accessible via the NFS protocol,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a setBackupStore --param backup_target_url="nfs://<host>:/<path>"
```

where:

- < `host` > is the address of the network storage server.
- < `path` > is the path to the backup network folder.

The path to the network folder for backup storage is specified.

Viewing the path to the network folder for backup storage

To view the current backup storage path,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a getBackupStore
```

The current backup storage path is displayed.

Configuring and managing the backup schedule

Kaspersky Next XDR Expert allows you to create backups according to a schedule, view the current schedule, pause or resume scheduled backups, and delete the schedule.

> You can configure only one backup schedule. Creating a new schedule will replace the existing one.

> Before using the scheduled backup, we recommend creating a backup manually to ensure that the network folder for storing backups is configured correctly and is accessible.

Configuring the backup schedule

To configure the backup schedule,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a updateSchedule \
  --param schedule='<schedule>' \
  --param schedule_paused=<schedule_is_paused> \
  --param backup_retain=<max_number_of_backups>
```

where:

- `<schedule>` is a line in the UNIX cron format that specifies the frequency of backups. For example, if you want to create a backup at 9:30 AM on the 1st and 15th of each month, and every Monday, use the following command:

  `./kdt invoke backup-operator -a updateSchedule --param schedule='30 9 1,15 * 1' --param schedule_paused=false --param backup_retain=5`

- `<schedule_is_paused>` is a parameter that specifies the state of the scheduled backup after the schedule is created:
  - `<schedule_is_paused>=false`—Scheduled backup is active and will run as configured.
  - `<schedule_is_paused>=true`—Scheduled backup is paused and will not run.
- `<max_number_of_backups>` is the maximum number of backups that can be stored in a network folder. The default value is 10. If the network folder already contains the maximum number of backups, the next backup will overwrite the oldest one. If you reduce the parameter value (for example, set `<max_number_of_backups>` to `3` when creating a backup manually or when configuring a backup schedule), KDT will delete all backups except the three most recent ones.

The backup schedule is configured.

View the details of the backup schedule

To view the details of the backup schedule,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a getBackupSchedule
```

The configured backup schedule details are displayed, including the schedule name shown in the `NAME` column.

Pausing the scheduled backup

To pause the scheduled backup,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a updateSchedule --param schedule_paused=true
```

The scheduled backup is paused.

Resuming the scheduled backup

To resume the scheduled backup,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a updateSchedule --param schedule_paused=false
```

The scheduled backup is resumed.

Creating a backup copy of Kaspersky Next XDR Expert data manually

If you need to save a backup without waiting for the next automatic backup, or ensure that the network folder for storing backups is configured correctly and is accessible, you can create a manual backup. During the backup process, the operation of Kaspersky Next XDR Expert components are temporarily suspended to ensure data consistency.

To create a backup manually,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a createBackup --param backup_retain=
<max_number_of_backups>
```

where `<max_number_of_backups>` is the maximum number of backups that can be stored in a network folder. The default value is 10. If the network folder already contains the maximum number of backups, the next backup will overwrite the oldest one. If you reduce the parameter value (for example, set `<max_number_of_backups>` to `3` when creating a backup manually or when configuring a backup schedule), KDT will delete all backups except the three most recent ones.

The created backup copy is saved to the configured network folder. KDT logs provide detailed information about the results of the backup operation, including its success or failure, the date and time of execution, and the name and location of the backup copy.

Viewing the backups of Kaspersky Next XDR Expert data

View the list of backups

To view the list of backups,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a listBackups
```

The list of backups is displayed, including the backup name shown in the `NAME` column and the unique backup identifier shown in the `BACKUPID` column.

View the details of the backup

To view the details of the backup,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a describeBackup --param backup_id='<backup_name>'
```

where `<backup_name>` is the backup name that is displayed in the `NAME` column in the list of the backups.

The details of the backup are displayed.

Restoring Kaspersky Next XDR Expert data from a backup copy

Before restoring data, ensure that you use the same version of Kaspersky Next XDR Expert in which the backup was created.

Backing up and restoring Kaspersky Next XDR Expert data is supported only if the DBMS is not included in the Kubernetes cluster. Configuration where the DBMS is installed into the Kubernetes cluster is used only to get acquainted with the solution, not to solve standard work tasks.

To restore the Kaspersky Next XDR Expert data from the backup copy,

On the administrator host, run the following command:

```
./kdt invoke backup-operator -a restoreFromBackup --param backup_id='<backup_id>'
```

where `<backup_id>` is the unique backup identifier that is displayed in the `BACKUPID` column in the list of the created backups.

Kaspersky Next XDR Expert data is restored from the backup. KDT outputs information about the success or failure of the restore operation in KDT logs.

Stopping the Kubernetes cluster nodes

You may need to stop the entire Kubernetes cluster or temporarily detach one of the nodes of the cluster for maintenance.

In a virtual environment, do not power off virtual machines that are hosting active Kubernetes cluster nodes.

To stop a multi-node Kubernetes cluster (multi-node deployment scheme):

1. Log in to a worker node and initiate graceful shut down. Repeat this process for all worker nodes.
2. Log in to the primary node and initiate graceful shut down.

To stop a single-node Kubernetes cluster (single-node deployment scheme):

Log in to the primary node and initiate graceful shut down.

Using certificates for public Kaspersky Next XDR Expert services

For working with public Kaspersky Next XDR Expert services, you can use self-signed or custom certificates. By default, Kaspersky Next XDR Expert uses self-signed certificates.

Certificates are required for the following Kaspersky Next XDR Expert public services:

- `<console_host>.<smp_domain>`—Access to the OSMP Console interface.
- `<ksc_host>.<smp_domain>`—Interaction with Administration Server.
- `<api_host>.<smp_domain>`—Access to the Kaspersky Next XDR Expert API.
- `<monitoring_host>.<smp_domain>`—Access to OSMP metrics.
- `<kuma_host>.<smp_domain>`—Access to the SIEM component.

FQDNs of public Kaspersky Next XDR Expert services consist of the host names and domain name specified in the configuration file. The list of addresses of public Kaspersky Next XDR Expert services, for which self-signed or custom certificates are defined during the deployment, is specified in the `pki_host_list` installation parameter.

A custom certificate must be specified as a file in the PEM format, and that contains the complete certificate chain (or only one certificate) and an unencrypted private key.

You can specify the intermediate certificate from your organization's private key infrastructure (PKI). Custom certificates for public Kaspersky Next XDR Expert services are issued from this custom intermediate certificate. Alternatively, you can specify leaf certificates for each of the public services. If leaf certificates are specified only for a part of the public services, then self-signed certificates are issued for the other public services.

For the `<console_host>.<smp_domain>`, `<kuma_host>.<smp_domain>`, `<monitoring_host>.<smp_domain>`, and `<api_host>.<smp_domain>` public services, you can specify custom certificates only before the deployment in the configuration file. Specify the `intermediate_bundle` and `intermediate_enabled` installation parameters to use the custom intermediate certificate. The validity period for the custom intermediate certificate must be 399 days or more.

If you want to use the custom leaf certificates to work with the public Kaspersky Next XDR Expert services, specify the corresponding `console_bundle`, `admsrv_bundle`, `kuma_bundle`, `monitoring_bundle`, and `api_bundle` installation parameters. Set the `intermediate_enabled` parameter to `false`, and do not specify the `intermediate_bundle` parameter and the `pki_host_list` parameter. The validity period for the custom leaf certificate must be 399 days or more.

For the <ksc_host>.<smp_domain> service, you can replace the issued Administration Server self-signed certificate with a custom certificate by using the klsetsrvcert utility.

> Automatic rotation of certificates is not supported. Take into account the validity term of the certificate, and then update the certificate when it expires.

To update custom certificates:

1. On the administrator host, export the current version of the configuration file.
2. In the exported configuration file, specify the path to a new custom intermediate certificate in the `intermediate_bundle` installation parameter. If you use the leaf custom certificates for each of the public services, specify the `console_bundle`, `admsrv_bundle`, `kuma_bundle`, `monitoring_bundle`, and `api_bundle` installation parameters.
3. Run the following command and specify the full path to the modified configuration file:

   ```
   ./kdt apply -i <full_path_to_configuration_file>
   ```

   Custom certificates are updated.

Calculation and changing of disk space for storing Administration Server data

Administration Server data includes the following objects:

- Information about assets (devices).
- Information about events logged on the Administration Server for the selected client device.
- Information about the domain in which the assets are included.
- Data of the Application Control component.
- Updates. The shared folder additionally requires at least 4 GB to store updates.
- Installation packages. If some installation packages are stored on the Administration Server, the shared folder will require an additional amount of free disk space equal to the total size of all of the available installation packages to be installed.
- Remote installation tasks. If remote installation tasks are present on the Administration Server, an additional amount of free disk space equal to the total size of all installation packages to be installed will be required.

Calculation of the minimum disk space for storing Administration Server data

The minimum disk space required for storing the Administration Server data can be estimated approximately by using the formula:

```
(724 * C + 0.15 * E + 0.17 * A + U), KB
```

where:

- C is the number of assets (devices).
- E is the number of events to store.
- A is the total number of domain objects:
  - Device accounts
  - User accounts
  - Accounts of security groups
  - Organizational units
- U is the size of updates (at least 4 GB).

If domain polling is disabled, A is considered to equal zero.

The formula calculates the disk space required for storing typical data from managed devices and the typical size of updates. The formula does not include the amount of disk space occupied by data that is independent of the number of managed devices for the Application Control component, installation packages, and remote installation tasks.

Changing of the disk space for storing the Administration Server data

The amount of free disk space allocated to store the Administration Server data is specified in the configuration file before the deployment of Kaspersky Next XDR Expert (the `ksc_state_size` parameter). Take into account the minimum disk space calculated by using the formula.

To check the disk space used to store the Administration Server data after the deployment of Kaspersky Next XDR Expert,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke ksc --action getPvSize
```

The amount of the required free disk space in gigabytes is displayed.

To change the disk space used to store the Administration Server data after the deployment of Kaspersky Next XDR Expert,

On the administrator host where the KDT utility is located, run the following command and specify the required free disk space in gigabytes (for example, "50Gi"):

```
./kdt invoke ksc --action setPvSize --param ksc_state_size='<new_disk_space_amount>Gi'
```

The amount of free disk space allocated to store the Administration Server data is changed.

Viewing and changing disk space for KUMA Core and the Metrics service

You can view the current amount of disk space allocated for the operation of KUMA Core and the Metrics service. If required, you can change the allocated disk space.

Viewing the currently allocated disk space

The commands for viewing the disk space do not affect the operation of the running services.

To view the disk space allocated for KUMA Core:

On the administrator host, run the following command:

```
./kdt invoke kuma --action getCorePvcSize
```

To view the disk space allocated for the Metrics service:

On the administrator host, run the following command:

```
./kdt invoke kuma --action getMetricsPvcSize
```

Changing the currently allocated disk space

The commands change the `core_disk_request` and the `metrics_disk_request` parameters. When you run a command, the corresponding service is stopped, the disk space is changed, and the service is restarted. The application updates the `core_disk_request` and `metrics_disk_request` parameters only if the commands are successfully completed.

You can only enlarge the current disk space. If you try to set a disk space value smaller than the current one, a command returns an error. In this case, the services do not stop.

If an error occurs after the services have already been stopped, the application tries to restart the services and keep the previous disk space value.

To change the disk space allocated for KUMA Core:

On the administrator host, run the following command:

```
./kdt invoke kuma --action setCorePvcSize --param core_disk_request=<value>
```

where `<value>` is new disk space size, for example `50Gi`.

To change the disk space allocated for the Metrics service:

On the administrator host, run the following command:

```
./kdt invoke kuma --action setMetricsPvcSize --param metrics_disk_request=<value>
```

where `<value>` is new disk space size, for example `50Gi`.

Rotation of Kubernetes cluster certificates

KDT allows you to rotate the Kubernetes cluster certificates that provide access to the Kubernetes cluster and to the infrastructure components of Kaspersky Next XDR Expert. Kubernetes cluster certificates are located on the target hosts inside the Kubernetes cluster.

To rotate Kubernetes cluster certificates manually,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke foreach --action RotateK0sConfig
```

New Kubernetes cluster certificates are generated. The default rotation period is 10 years.

When updating Bootstrap, Kubernetes cluster certificates are updated automatically.

Rotation of Registry access credentials

KDT allows you to rotate Registry access credentials in accordance with your organization's information security requirements.

To rotate Registry access credentials manually,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke bootstrap --action RotateRegistryCreds
```

Registry access credentials are updated.

When updating Bootstrap, Registry access credentials are updated automatically.

Adding hosts for installing the additional KUMA services

If you need to expand the storage, or add new collectors and correlators for the increased flow of events, you can add additional hosts for installation of the KUMA services.

You must specify the parameters of the additional hosts in the expand.inventory.yml file. This file is located in the distribution package with the transport archive, KDT, the configuration file, and other files. In the expand.inventory.yml file, you can specify several additional hosts for collectors, correlators, and storages at once. Ensure that hardware, software, and installation requirements for the selected hosts are met.

To prepare the required infrastructure on the hosts specified in the expand.inventory.yml file, you need to create the service directories to which the files that are required for the service installation are added. To prepare the infrastructure, run the following command and specify the expand.inventory.yml file:

```
./kdt invoke kuma --action addHosts --param host_inventory='<path_to_inventory_file>'
```

On the hosts specified in the expand.inventory.yml file, the service directories to which the files that are required for the service installation are added.

Sample of the additional KUMA inventory file for installation of the KUMA services (the expand.inventory.yml file)

```
all:
  vars:
    ansible_connection: local
    ansible_user: nonroot
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
  children:
    kuma_utils:
    kuma_collector:
      hosts:
        kuma1.example.com:
            ansible_host: 0.0.0.0
        kuma2.example.com:
            ansible_host: 0.0.0.0
    kuma_correlator:
      hosts:
        kuma3.example.com:
            ansible_host: 0.0.0.0
        kuma4.example.com:
            ansible_host: 0.0.0.0
    kuma_storage:
      hosts:
        kuma5.example.com:
            ansible_host: 0.0.0.0
        kuma6.example.com:
            ansible_host: 0.0.0.0
```

Adding an additional storage, collector, or correlator

You can add an additional storage cluster, collector, or correlator to your existing infrastructure. If you want to add several services, it is recommended to install them in the following order: storages, collectors, and correlators.

To add an additional storage cluster, collector, or correlator:

1. In the main menu of OSMP Console, go to **Settings** → **KUMA**.

2. In KUMA section of OSMP Console, create a resource set for each KUMA service (storages, collectors, and correlators) that you want to install on the prepared hosts.

3. Create services for storages, collectors, and correlators.

4. Obtain the service identifiers to bind the created resource sets and the KUMA services:

   a. In the KUMA section main menu, go to **Resources** → **Active services**.

   b. Select the required KUMA service, and then click the **Copy ID** button.

5. Install the KUMA services on each prepared host listed in the kuma_storage, kuma_collector, and kuma_correlator sections of the expand.inventory.yml inventory file. On each machine, in the installation command, specify the service ID corresponding to the host. Run the corresponding commands to install the KUMA services:

   - Installation command for the storage:

     ```
     sudo /opt/kaspersky/kuma/kuma storage --core https://<KUMA Core server FQDN>:7210 --
     id <service ID copied from the KUMA section> --install
     ```

   - Installation command for the collector:

     ```
     sudo /opt/kaspersky/kuma/kuma collector --core https://<KUMA Core server FQDN>:7210
     --id <service ID copied from the KUMA section> --api.port <port used for
     communication with the installed component>
     ```

   - Installation command for the correlator:

     ```
     sudo /opt/kaspersky/kuma/kuma correlator --core https://<KUMA Core server FQDN>:7210
     --id <service ID copied from the KUMA section> --api.port <port used for
     communication with the installed component> --install
     ```

     The collector and correlator installation commands are automatically generated on the **Setup validation** tab of the Installation Wizard, and the port used for communication is added to the command automatically. Use the generated commands to install the collector and correlator on the hosts. This will allow you to make sure that the ports for communication with the services specified in the command are available.

   The FQDN of KUMA Core is `<kuma_host>.<smp_domain>`.

   The port that is used for connection to KUMA Core cannot be changed. By default, port 7210 is used.

The additional KUMA services are installed.

Adding hosts to an existing storage

You can expand an existing storage (storage cluster) by adding hosts as new storage cluster nodes.

To add hosts to an existing storage:

1. In the main menu of OSMP Console, go to **Settings** → **KUMA**.

2. Add new nodes to the storage cluster. To do this, edit the settings of the existing storage cluster:

   a. In the **Resources** → **Storages** section, select an existing storage, and then open the storage for editing.

   b. In the **ClickHouse cluster nodes** section, click **Add nodes**, and then specify roles in the fields for the new node. Specify the corresponding host domain names from the kuma_storage section of the expand.inventory.yml file, and then specify the roles for the new nodes.

   c. Save changes.

   You do not need to create a separate storage because you are adding servers to an existing storage cluster.

3. Create storage services for each added storage cluster node in the KUMA section of OSMP Console, and then bind the services to the storage cluster.

4. Obtain the storage service identifiers for each prepared host to install the KUMA services:

   a. In the KUMA section of OSMP Console, go to **Resources** → **Active services**.

   b. Select the required KUMA service, and then click the **Copy ID** button.

5. Install the storage service on each prepared host listed in the kuma_storage section of the expand.inventory.yml inventory file. On each machine, in the installation command, specify the service ID corresponding to the host. Run the following command to install the storage service:

   ```
   sudo /opt/kaspersky/kuma/kuma storage --core https://<KUMA Core server FQDN>:7210 --id
   <service ID copied from the KUMA section> --install
   ```

   The FQDN of the KUMA Core is `<kuma_host>.<smp_domain>`.

   The port that is used for connection to KUMA Core cannot be changed. By default, port 7210 is used.

   The additional hosts are added to the storage cluster.

Specify the added hosts in the distributed.inventory.yml inventory file so that it has up-to-date information in case of a KUMA components update.

Replacing a host that uses KUMA storage

To replace a host that uses KUMA storage with another one:

1. Fill in the expand.inventory.yml file, specifying the parameters of the host you want to replace.

2. Run the following command, specifying the expand.inventory.yml file to remove the host:

   ```
   ./kdt invoke kuma --action removeHosts --param
   host_inventory='<path_to_inventory_file>'
   ```

3. Fill in the expand.inventory.yml file, specifying the parameters of the new host that you want to replace the previous host, and then run the following command:

   ```
   ./kdt invoke kuma --action addHosts --param host_inventory='<path_to_inventory_file>'
   ```

4. Follow steps 2–6 of the instruction for adding new hosts for KUMA services to add a new host with the KUMA storage.

The host with the KUMA storage is replaced with another one.

If your storage configuration includes a shard containing two replicas, and you replaced the second replica host with a new one by using the steps described above, then you may receive an error when installing a new replica. In this case, the new replica will not work.

To fix an error when adding a new replica of a shard:

1. On another host with a replica of the same shard that owns the incorrectly added replica, launch the ClickHouse client by using the command:

   ```
   /opt/kaspersky/kuma/clickhouse/bin/client.sh
   ```

   If this host is unavailable, run the client on any other host with a replica included in the same storage cluster.

2. Run the command to remove the data about the host you wanted to replace.

   - If the host with a replica of the same shard that owns the incorrectly added replica is available, run the following command:

     ```
     SYSTEM DROP REPLICA '<replica number of read-only node>' FROM TABLE
     kuma.events_local_v2
     ```

   - If you are using another storage cluster host with a replica, run the following command:

     ```
     SYSTEM DROP REPLICA '<replica number of read-only node>' FROM ZKPATH
     '/clickhouse/tables/kuma/<shard number of read-only node>/kuma/events_local_v2
     ```

3. Run the following command to restore the operation of the added host with a replica:

   ```
   SYSTEM RESTORE REPLICA kuma.events_local_v2
   ```

Operability of the added host with a replica is restored.

Working with Kaspersky Next XDR Expert deployed on virtual machines

If you have deployed Kaspersky Next XDR Expert on virtual machines, we do not recommend taking snapshots or migrating these virtual machines by using a hypervisor while the solution is running. In this case, Kaspersky Next XDR Expert will not function properly. To ensure proper functioning of Kaspersky Next XDR Expert, stop all virtual machines within the Kubernetes cluster before creating snapshots or performing the migration.

Demonstration deployment of Kaspersky Next XDR Expert

If you want to get acquainted with Kaspersky Next XDR Expert, you can perform a demonstration deployment of the solution.

Use the demonstration deployment of Kaspersky Next XDR Expert (in which the DBMS is installed into the Kubernetes cluster) only to get acquainted the solution. We do not recommend using this approach to solve standard work tasks. For standard usage, perform a multi-node or single-node deployment.

In the demonstration deployment, all Kaspersky Next XDR Expert components are installed on a single node or on multiple nodes of the Kubernetes cluster.

In the scenario below, the demonstration deployment on a single node is described. Single-node deployment requires less resources and is therefore more often used for demonstration deployment. In the demonstration deployment on a single node, the Kubernetes cluster, Kaspersky Next XDR Expert components, and the DBMS are installed on one target host. Also you need the administrator host to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert, as well as the target host for installing the KUMA services.

In the single-node configuration, you need at least three hosts:

- 1 administrator host
- 1 target host for installing the Kubernetes cluster, the Kaspersky Next XDR Expert components, and the DBMS
- 1 KUMA target host for installing the KUMA services

The main difference between a demonstration deployment and a standard one is the DBMS location. In a standard deployment (multi-node or single-node), the DBMS is installed on a host located outside the cluster. In a demonstration deployment, the DBMS is installed on the target host inside the cluster.

Before you deploy Open Single Management Platform and Kaspersky Next XDR Expert components, we recommend reading the Hardening Guide.

Following this scenario, you can prepare the infrastructure for the demonstration deployment, prepare the configuration file, and deploy the solution by using the Kaspersky Deployment Toolkit utility (hereinafter referred to as KDT).

The demonstration deployment scenario proceeds in stages:

1. **Downloading the distribution package with the Kaspersky Next XDR Expert components**

   The distribution package contains the following components:

   - Transport archive with the Kaspersky Next XDR Expert components and End User License Agreements for Kaspersky Next XDR Expert and KDT
   - Archive with the KDT utility, and templates of the configuration file and KUMA inventory file

2. **Preparing the administrator and target hosts**

   Prepare the selected administrator host and target hosts for a demonstration deployment of Kaspersky Next XDR Expert.

   How-to instruction: Preparing the administrator and target hosts

3. **Preparing the KUMA hosts**

   Prepare the KUMA target hosts for the installation of the KUMA services (collectors, correlators, and storages).

   How-to instruction: Preparing the hosts for installation of the KUMA services

4. **Preparing the KUMA inventory file for installation of the KUMA services**

   Prepare the KUMA inventory file in the YAML format. The KUMA inventory file contains parameters for installation of the KUMA services.

   How-to instruction: Preparing the KUMA inventory file

5. **Preparing the configuration file**

   Prepare the configuration file in the YAML format. The configuration file contains the target host parameters and a set of installation parameters of the Kaspersky Next XDR Expert components.

   How-to instructions: Specifying the installation parameters

6. **Deployment of Kaspersky Next XDR Expert**

   Deploy Kaspersky Next XDR Expert by using KDT. KDT automatically deploys the Kubernetes cluster within which the Kaspersky Next XDR Expert components and other infrastructure components are installed.

   How-to instruction: Installing Kaspersky Next XDR Expert

7. **Installing the KUMA services**

   Install the KUMA services (collectors, correlators, and storages) on the prepared KUMA target hosts that are located outside the Kubernetes cluster.

   How-to instruction: Installing KUMA services

8. **Configuring integration with Kaspersky Anti Targeted Attack Platform**

   Install Central Node to receive telemetry from Kaspersky Anti Targeted Attack Platform, and then configure integration between Kaspersky Next XDR Expert and KATA/KEDR to manage threat response actions on assets connected to Kaspersky Endpoint Detection and Response servers.

   If necessary, you can install multiple Central Node components to use them independently of each other or to combine them for centralized management in the distributed solution mode. To combine multiple Central Node components, you have to organize the servers with the components into a hierarchy.

   When configuring the Central Node servers, you have to specify the minimum possible value in the **Storage** field, to avoid duplication of data between the Kaspersky Next XDR Expert and KEDR databases.

Preparing the administrator and target hosts

Preparing for a demonstration deployment includes configuring the administrator and target hosts. In the demonstration deployment on a single node, the Kubernetes cluster, Kaspersky Next XDR Expert components, and the DBMS are installed on one target host. After preparing the target host and specifying the configuration file, you will be able to deploy Kaspersky Next XDR Expert on the target host by using KDT.

Use the demonstration deployment of Kaspersky Next XDR Expert (in which the DBMS is installed into the Kubernetes cluster) only to get acquainted the solution. We do not recommend using this approach to solve standard work tasks. For standard usage, perform a multi-node or single-node deployment.

Preparing the administrator host

You first need to prepare a device that will act as the administrator host from which KDT will launch. This host can be either included in the Kubernetes cluster that is created by KDT during the deployment or not. If the administrator host is not included in the cluster, it will be used only to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert. If the administrator host is included in the cluster, it will also act as a target host that is used for operation of Kaspersky Next XDR Expert components. In this case, only one host will be used for deployment and operation of the solution.

To prepare the administrator host:

1. Make sure that the hardware and software on the administrator host meet the requirements for KDT.

2. Allocate at least 10 GB of free space in the temporary files directory (`/tmp`) for KDT. If you do not have enough free space in this directory, run the following command to specify the path to another directory:

   ```
   export TMPDIR=<new_directory>/tmp
   ```

3. Install the package for Docker version 23 or later, and then perform the post-installation steps to configure the administration host for proper functioning with Docker.

   Do not install unofficial distributions of Docker packages from the operating system maintainer repositories.

4. For the administrator host that will be included in the cluster, perform additional preparatory steps:

   a. Since the device will act as the administrator and target host, make sure that it meets the requirements for the single-node deployment.

   b. Make sure that the cgroup v2 technology is supported on the administrator host.

   a. Since the device will act as the administrator and target host, make sure that it meets the requirements for the single-node deployment (because the demonstration and single-node deployment have a similar deployment scheme).

   b. Make sure that the cgroup v2 technology is supported on the administrator host.

      The cgroup v2 technology is supported for the Linux kernel version 2.6.24 or later.

   c. Install the uidmap package on the administrator host.

Check that the /etc/subgid and /etc/subuid files contain the user account under which KDT will be launched. To do this, you can run the following command:

```
getsubids USER
```

If this command does not return a result, you must add the user account to the /etc/subgid and /etc/subuid files manually in the following format:

```
<username>:<min_subid>:<range_length>
```

where

- `<username>`—Username of the account under which KDT will be launched.
- `<min_subid>`—Minimum subuid value.
- `<range_length>`—Number of subuids allocated for the user `<username>`.

Preparing the target hosts

The target host is a physical or virtual machine that is used to deploy Kaspersky Next XDR Expert and included in the Kubernetes cluster. The target host manages the Kubernetes cluster, stores metadata, as well as the Kaspersky Next XDR Expert components work on this host. A minimum cluster configuration for the demonstration deployment includes one target host, which acts as the primary and worker nodes. On this primary worker node, the Kubernetes cluster, the DBMS, and Kaspersky Next XDR Expert components are installed.

The KUMA services (collectors, correlators, and storages) can be installed entirely inside the cluster on the primary worker node, entirely on separate hosts outside the Kubernetes cluster, or partially inside and partially outside the Kubernetes cluster.

If you want to run the Kaspersky Next XDR Expert deployment from the target host, you must prepare this host as the administrator host, as described in the previous procedure, and then perform the preparing for the target host.

To prepare the target host:

1. Make sure that the hardware and software on the target host meet the requirements for the single-node deployment (because the demonstration and single-node deployment have a similar deployment scheme).

   If KUMA services are installed inside the cluster, the target host must meet the hardware, software, and installation requirements for installing the KUMA services.

   For proper functioning of Kaspersky Next XDR Expert, the Linux kernel version must be 5.15.0.107 or later on the target host with the Ubuntu family operating systems.

   Do not install Docker on the target host unless the target host will be used as the administrator host. KDT will install all necessary software and dependencies during the deployment.

2. Ensure that the filesystem mounted at `/var` has at least 360 GB of free space on the target host.

3. Install the sudo package, if this package is not already installed. For Debian family operating systems, install the UFW package.

4. Configure the /etc/environment file. If your organization's infrastructure uses a proxy server to access the internet, you also need to connect the target host to the internet.

5. If the primary worker node has the UFW configuration, allow IP forwarding. In the `/etc/default/ufw` file, set `DEFAULT_FORWARD_POLICY` to `ACCEPT`.

6. Provide access to the package repository. This repository stores the following packages required for Kaspersky Next XDR Expert:

   - nfs-common
   - tar
   - open-iscsi

   KDT will try to install these packages during the deployment from the package repository. You can also install these packages manually.

7. Ensure that the python3-apt and curl packages are installed on the primary worker node.

   The python3-apt and curl packages are not installed during the deployment from the package repository by using KDT. You must install these packages manually, if they are not already installed.

8. Reserve static IP addresses for the target host and for the Kubernetes cluster gateway.

   The Kubernetes cluster gateway is intended for connecting to the Kaspersky Next XDR Expert components installed inside the Kubernetes cluster.

   Since you install the DBMS inside the Kubernetes cluster, the gateway IP address is an IP range (for example, 192.168.0.1—192.168.0.2). The first IP address of the range is the address of the Kaspersky Next XDR Expert services (excluding the DBMS IP address), and the second IP address of the range is the IP address of the DBMS.

   Make sure that the target host and the Kubernetes cluster gateway are located in the same broadcast domain.

9. On your DNS server, register the service FQDNs to connect to the Kaspersky Next XDR Expert services.

   By default, the Kaspersky Next XDR Expert services are available at the following addresses:

   - <console_host>.<smp_domain>—Access to the OSMP Console interface.
   - <ksc_host>.<smp_domain>—Interaction with Administration Server.
   - <kuma_host>.<smp_domain>—Access to the SIEM component.
   - <api_host>.<smp_domain>—Access to the Kaspersky Next XDR Expert API.
   - <monitoring_host>.<smp_domain>—Access to OSMP metrics.
   - <psql_host>.<smp_domain>—Interaction with the DBMS (PostgreSQL).
   - <xdr_edr_ingress_host>.<smp_domain>—Interaction with Agent server.

     Where <console_host>, <ksc_host>, <kuma_host>, <api_host>, <monitoring_host>, <psql_host>, and <xdr_edr_ingress_host> are service host names, <smp_domain> is a service domain name. These parameters are parts of the service FQDNs, which you can specify in the configuration file. If you do not specify custom values of service host names, the default values are used: `console_host`—`"console"`, `ksc_host`—`"admsrv"`, `kuma_host`—`"kuma"`, `api_host`—`"api"`, `monitoring_host`—`"monitoring"`, `psql_host`—`"psql"`, `"xdr_edr_ingress_host"`—`"xdr_edr_ingress"`.

The listed service FQDNs must be resolved to the IP address of the Kubernetes cluster. For example, if the Kubernetes cluster address is 10.80.23.182, service FQDNs are resolved to the IP address as follows:

- <console_host>.<smp_domain>—192.168.0.1
- <ksc_host>.<smp_domain>—192.168.0.1
- <kuma_host>.<smp_domain>—192.168.0.1
- <api_host>.<smp_domain>—192.168.0.1
- <psql_host>.<smp_domain>—192.168.0.2
- <xdr_edr_ingress_host>.<smp_domain>—192.168.0.1

10. Create the user accounts that will be used for the Kaspersky Next XDR Expert deployment.

    These accounts are used for the SSH connection and must be able to elevate privileges (sudo) without entering a password. To do this, add the created user accounts to the `/etc/sudoers` file.

    View details

    To add the created user accounts to the `/etc/sudoers` file:

    1. Run the following commands:

       ```
       sudo useradd -m -s /bin/bash <user_name>
       sudo usermod -aG sudo <user_name>
       ```

    2. Open the `/etc/sudoers` file by using the following command:

       ```
       sudo visudo
       ```

    3. Add the following line to the end of the section `# Allow members of group sudo to execute any command`:

       ```
       <user_name> ALL=(ALL) NOPASSWD: ALL
       ```

       where `<user_name>` is the name of the created user.

    4. Save and close the `/etc/sudoers` file.

11. Configure the SSH connection between the administrator and target hosts:

    a. On the administrator host, generate SSH keys by using the ssh-keygen utility without a passphrase.

    b. Copy the public key to the target host (for example, to the `/home/<user_name>/.ssh` directory) by using the ssh-copy-id utility.

       If you use the target host as the administrator host, you must copy the public key to it, too.

12. For proper function of the Kaspersky Next XDR Expert components, open the required ports on the firewall of the administrator and target hosts, if necessary.

13. Configure time synchronization over Network Time Protocol (NTP) on the administrator and target hosts.

14. If necessary, prepare custom certificates for working with Kaspersky Next XDR Expert public services.

    You can use one intermediate certificate that is issued off the organization's root certificate or leaf certificates for each of the services. The prepared custom certificates will be used instead of self-signed certificates.

Specifying the installation parameters

The configuration file used to deploy Kaspersky Next XDR Expert on a single node contains the installation parameters that are required both for the multi-node and single-node deployment. Also, this configuration file contains parameters specific only for the single-node deployment (`openbao_ha_mode`, `openbao_standalone`, and `default_class_replica_count`). The template of the configuration file (singlenode.smp_param.yaml.template) is located in the distribution package in the archive with the KDT utility. This configuration file can be used for a demonstration installation.

Use the demonstration deployment of Kaspersky Next XDR Expert (in which the DBMS is installed into the Kubernetes cluster) only to get acquainted the solution. We do not recommend using this approach to solve standard work tasks. For standard usage, perform a multi-node or single-node deployment.

Not all of the parameters listed below are included in the configuration file template. This template contains only those parameters that must be specified before Kaspersky Next XDR Expert deployment. The remaining parameters are set to default values, and they are not included in the template. You can manually add these parameters to the configuration file to override its values. Parameter values must follow the rules described below. During the Kaspersky Next XDR Expert deployment, KDT checks whether the required installation parameters are specified correctly. If the required parameters in the configuration file are missing or invalid, the deployment will be interrupted.

For correct function of KDT with the configuration file, add an empty line at the end of the file.

The `nodes` section of the configuration file contains the target host parameters that are listed in the table below.

Nodes section

Parameter name	Required	Description

`desc`	Yes	The node name.
		The value must comply with the following rules:
		• The node name must be 1 to 63 characters long.
		• The node name can contain the following characters:
		  • Uppercase ASCII letters (A–Z)
		  • Lowercase ASCII letters (a–z)
		  • Numbers (0–9)
		  • Hyphen (-)
		All cluster nodes must have unique names. This requirement is one of the prerequisites for the installation of Kaspersky Next XDR Expert and is verified by KDT.

`type`	Yes	The node type. For the target host, set the `type` parameter to `primary-worker` to enable the **demonstration deployment on a single node**. In this case, the target host will act as the primary and worker nodes.

Parameter name	Required	Description
`host`	Yes	The IP address of the node. The primary worker node must be included in the same subnet as the Kubernetes cluster gateway. The node requires IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on the node and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks.
`kind`	No	The node kind that specifies the Kaspersky Next XDR Expert component that will be installed on this node. For the **single-node demonstration deployment**, leave this parameter empty, because all components will be installed on a single node.

`user`	Yes	The user name of the account created on the target host and used for connection to the node by KDT.
		The value must comply with the following rules:
		• The user name must be 1 to 31 characters long.
		• The user name can contain the following characters:
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Underscore (_), hyphen (-)

`password`	No	The password for connecting to the node. Can be used instead of the SSH key specified by the `ssh_pk` parameter.

Other installation parameters are listed in the `parameters` section of the configuration file and are described in the table below.

Parameters section

Parameter name	Required	Description
`nwc-language`	Yes	The language of the OSMP Console interface is specified by default. After installation, you can change the OSMP Console language. Possible parameter values: • `enUS` (default value) • `ruRu` The default value is used if you do not specify this parameter in the configuration file.
`ingress_ip`	Yes	The reserved static IPv4 address of the Kubernetes cluster gateway. The Kubernetes cluster gateway is intended for connecting to the Kaspersky Next XDR Expert components installed inside the Kubernetes cluster. The gateway must be included in the same subnet as the primary worker node. The Kubernetes cluster gateway requires IPv4 addressing. If your environment requires IPv6 connectivity, configure the operating systems on the node and the Kubernetes cluster gateway to route IPv6 traffic through an IPv6 gateway (NAT64) or an IPv6 tunnel. This configuration keeps the cluster running on IPv4 while providing interoperability with IPv6 networks. The value of this parameter cannot be the IP address of the same host on which the deployment is being performed.
`pg_ip`	Yes	The IP address of the DBMS. For the **demonstration deployment**, when you install the DBMS inside the cluster, the DBMS IP address does not match the cluster gateway IP address and is set by using the `pg_ip` parameter.
`ssh_pk`	Yes	The path to the private part of the SSH key located on the administrator host and used for connection to the primary worker node and nodes with the KUMA services (collectors, correlators, and storages) by using KDT. The parameter value must be a Linux file path or Base64-encoded file content.

Parameter name	Required	Description
`admin_password`	Yes	The `admin_password` parameter specifies the password of the Kaspersky Next XDR Expert user account that will be created by KDT during the installation. The default user name of this account is "admin". The Main administrator role is assigned to this user account. The password must comply with the following rules: • The user password cannot have fewer than 8 or more than 256 characters. • The password must contain characters from at least three of the groups listed below: • Uppercase letters (A–Z) • Lowercase letters (a–z) • Numbers (0–9) • Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;) • The password must not contain any whitespaces, or the ".@" combination. When you specify the `admin_password` parameter value manually (not by the Configuration wizard), make sure that this value meets the YAML standard requirements for values in strings: • The parameter value containing special characters must be enclosed in single quotes. • Any single quote ' inside the parameter value must be doubled to escape this single quote. Example: the user account password `Any_pass%1234'5678"90` must be specified as the value `'Any_pass%1234''5678"90'` of the `admin_password` parameter.
`low_resources`	Yes	The parameter that indicates that Kaspersky Next XDR Expert is installed on the target host with limited computing resources. For the **demonstration deployment on a single node**, set the `low_resources` parameter to `true` so that Kaspersky Next XDR Expert components will require less memory and CPU resources. Also, if you enable this parameter, 4 GB of free disk space will be allocated to install KUMA Core on the target host.
`openbao_ha_mode`	Yes	The parameter that indicates whether to run the secret storage in the High Availability (HA) mode. For the **demonstration deployment on a single node**, set the `openbao_ha_mode` parameter to `false`.
`openbao_standalone`	Yes	The parameter that indicates whether to run the secret storage in the standalone mode. For the **demonstration deployment on a single node**, set the `openbao_standalone` parameter value to `true`.
`default_class_replica_count`	Yes	The number of disk volumes that are used to store the service data of Kaspersky Next XDR Expert components and KDT. The default value is `3`. For the **demonstration deployment on a single node**, set the `default_class_replica_count` parameter value to `1`.
`core_disk_request`	Yes	The parameter that specifies the amount of disk space for the operation of KUMA Core. If you do not specify the `core_disk_request` parameter and the `low_resources` parameter is set to `false` or is not specified, the default amount of disk space for the operation of KUMA Core is 100 GB. If you do not specify the `core_disk_request` parameter and the `low_resources` parameter is set to `true`, the default amount of disk space for the operation of KUMA Core is 4 GB. You can change the `core_disk_request` parameter later, after the application deployment.
`inventory`	Yes	The path to the KUMA inventory file located on the administrator host. The inventory file contains installation parameters for deployment of the KUMA services that are not included in the Kubernetes cluster. The parameter value must be a Linux file path or Base64-encoded file content. If all KUMA services will be installed inside the cluster, set the `inventory` parameter to `/dev/null`.

Parameter name	Required	Description
host_inventory	No	The path to the additional KUMA inventory file located on the administrator host. This file contains the installation parameters used to partially add or remove hosts with the KUMA services. The parameter value must be a Linux file path or Base64-encoded file content. If you perform an initial deployment of Kaspersky Next XDR Expert or run a custom action that requires configuration file, leave the default parameter value (/dev/null).
license	Yes	The path to the KUMA license key located on the administrator host. The parameter value must be a Linux file path or Base64-encoded file content. You can upload the license key or specify the activation code once you log in to OSMP after deployment is complete.
nwc_host flow_host hydra_host login_host ksc_host console_host api_host kuma_host psql_host monitoring_host	Yes	The host name that is used in the FQDNs of the public Kaspersky Next XDR Expert services. The service host name and domain name (the smp_domain parameter value) are parts of the service FQDN. Default values of the parameters: • nwc_host—"console" • flow_host—"console" • hydra_host—"console" • login_host—"console" • ksc_host—"admsrv" • console_host—"console" • api_host—"api" • kuma_host—"kuma" • psql_host—"psql" (only for the **demonstration deployment)** • monitoring_host—"monitoring"
smp_domain	Yes	The domain name that is used in the FQDNs of the public Kaspersky Next XDR Expert services. The parameter value must meet the requirements for second-level domain naming. The service host name and domain name are parts of the service FQDN. For example, if the value of the console_host variable is console, and the value of the smp_domain variable is smp.local, then the full name of the service that provides access to the OSMP Console is console.smp.local.
pki_host_list	No	The list of host names of the public Kaspersky Next XDR Expert services for which a self-signed or custom certificate is to be generated. The parameter value must be a list of host names, separated by spaces.
intermediate_enabled	No	The parameter that indicates whether to use the custom intermediate certificate instead of the self-signed certificates for the public Kaspersky Next XDR Expert services. Possible parameter values: • true—Use custom intermediate certificate (default value). • false—Use self-signed certificates.
intermediate_bundle	No	The path to the custom intermediate certificate used to work with public Kaspersky Next XDR Expert services. Specify this parameter if the intermediate_enabled parameter is set to true. The parameter value must be a Linux file path or Base64-encoded file content. The validity period for the custom intermediate certificate must be 399 days or more.
admsrv_bundle api_bundle console_bundle psql_bundle kuma_bundle monitoring_bundle	No	The paths to the custom leaf certificates used to work with the corresponding public Kaspersky Next XDR Expert services: <ksc_host>.<smp_domain>, <api_host>.<smp_domain>, <console_host>.<smp_domain>, <kuma_host>.<smp_domain>, <monitoring_host>.<smp_domain>, and <psql_host>.<smp_domain>. The parameter values must be a Linux file path or Base64-encoded file content. Specify the psql_bundle parameter only if you perform the **demonstration deployment** and install the DBMS inside the Kubernetes cluster. If you want to specify the custom leaf certificates, set the intermediate_enabled parameter to false and do not specify the intermediate_bundle parameter.
encrypt_secret sign_secret	Yes	The names of the secret files that are stored in the Kubernetes cluster. These names contain the domain name, which must match the smp_domain parameter value.

Parameter name	Required	Description
`ksc_state_size`	Yes	The amount of free disk space allocated to store the Administration Server data (updates, installation packages, and other internal service data). Measured in gigabytes, specified as "<amount>Gi". The required amount of free disk space depends on the number of managed devices and other parameters, and can be calculated. The default value is 30 GB. This value is used if you do not specify this parameter in the configuration file.
`prometheus_size`	Yes	The amount of free disk space allocated to store metrics. Measured in gigabytes, specified as "<amount>GB". The minimum recommended and default value is 5 GB. The default value is used if you do not specify this parameter in the configuration file.

`grafana_admin_user`	No	The user name of the account used to view OSMP metrics through the Grafana tool.
		The value must comply with the following rules:
		• The user name must be 1 to 30 characters long.
		• The user name can contain the following characters:
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Special characters (_ - .)
		• The user name must start with a letter.
		• The user name must not contain either whitespaces or the "@" symbol.

`grafana_admin_password`	No	The password of the account used to view OSMP metrics through the Grafana tool.
		The password must comply with the following rules:
		• The user password cannot have fewer than 8 or more than 256 characters.
		• The password must contain characters from at least three of the groups listed below:
		  • Uppercase letters (A–Z)
		  • Lowercase letters (a–z)
		  • Numbers (0–9)
		  • Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)
		• The password must not contain either whitespaces or ".@".

`loki_size`	Yes	The amount of free disk space allocated to store OSMP logs. Measured in gigabytes, specified as "<amount>Gi". The minimum recommended and default value is 20 GB. The default value is used if you do not specify this parameter in the configuration file.
`loki_retention_period`	Yes	The storage period of OSMP logs after which logs are automatically removed. The default value is 72 hours (set the parameter value in the configuration file as "<time in hours>h". For example, "72h"). The default value is used if you do not specify this parameter in the configuration file.
`file_storage_cp`	No	The amount of free disk space allocated to store data of the component for working with response actions. Measured in gigabytes, specified as "<amount>Gi". The minimum recommended value is 20 GB.
`psql_tls_off`	No	The parameter that indicates whether to encrypt the traffic between the Kaspersky Next XDR Expert components and the DBMS by using the TLS protocol. If the DBMS is installed outside the cluster, TLS encryption is disabled by default. For the **demonstration deployment**, if the DBMS is installed inside the cluster, TLS encryption must be disabled (`psql_tls_off` is set to `true`). Possible parameter values: • `true` —Do not encrypt the traffic (default value). • `false` —Encrypt the traffic.

Parameter name	Required	Description
`proxy_enabled`	No	The parameter that indicates whether to use the proxy server to connect the Kaspersky Next XDR Expert components to the internet. If the host on which Kaspersky Next XDR Expert is installed has internet access, you can also provide internet access for the operation of Kaspersky Next XDR Expert components (for example, Administration Server) and for specific integrations, both Kaspersky and third-party. To establish the proxy connection, you must also specify the proxy server parameters in the Administration Server properties. Possible parameter values: • `true` —Proxy server is used. • `false` —Proxy server is not used (default value).
`proxy_addresses`	No	The IP address of the proxy server. If the proxy server uses multiple IP addresses, specify these addresses separated by a space (for example, "`0.0.0.0 0.0.0.1 0.0.0.2`"). Specify this parameter if the `proxy_enabled` parameter is set to `true`.
`proxy_port`	No	The number of the port through which the proxy connection will be established. Specify this parameter if the `proxy_enabled` parameter is set to `true`.
`trace_level`	No	The verbosity level of Administration Server trace files. Traces are written during the Administration Server operation. The default value is 0, no traces are written. Possible parameter values: 0–5. The higher the parameter value, the more detailed the trace details.
`ansible_extra_flags`	No	The verbosity level of logs of the KUMA Core and KUMA services deployment that is performed by KDT. Possible parameter values: • `-v` • `-vv` • `-vvv` • `-vvvv` As the number of "v" letters in the flag increases, logs become more detailed. If this parameter is not specified in the configuration file, the standard component installation logs are saved.
`incident_attachments_max_count_limit`	No	The number of files that you can attach to the incident. The default value is `100`.
`incident_attachments_max_size_limit`	No	The total size of files attached to the incident. Measured in bytes. Specified without units of measurement. The default value is `26214400`.
`ignore_precheck`	No	The parameter indicating whether to check the hardware, software, and network configuration of the Kubernetes cluster nodes for compliance with the prerequisites for installing the solution before the deployment. Possible parameter values: • `true` —Skip the pre-checks. • `false` —Perform the pre-checks (default value).
`grafana_smtp_host`	No	SMTP server address for Grafana, used to send email notifications about Kubernetes cluster health events. Specify this parameter in the `<host>:<port>` format, for example, `localhost:25`.
`grafana_smtp_user`	No	User name for authentication on the SMTP server.
`grafana_smtp_password`	No	Password for authentication on the SMTP server.
`grafana_smtp_bundle`	No	File containing the X.509 certificate chain used for verifying TLS connection to the SMTP server.
`grafana_smtp_skip_verify`	No	The parameter that indicates whether to skip TLS certificate verification of the SMTP server. Possible parameter values: • `true` —Skip TLS certificate verification. • `false` —Perform TLS certificate verification (default value).
`grafana_smtp_from_address`	No	Email address of the sender, which will appear in the From field when sending email notifications.
`grafana_smtp_from_name`	No	Display name of the sender in email notifications.
`grafana_smtp_ehlo_identity`	No	Identifier used in the EHLO command when connecting to the SMTP server.

Parameter name	Required	Description
`grafana_starttls_policy`	No	Policy for using STARTTLS when connecting to the SMTP server. Possible parameter values: • `OpportunisticStartTLS` —Use STARTTLS if the SMTP server supports it; otherwise, connect to the SMPT server without encryption. • `MandatoryStartTLS` —Require STARTTLS; fail if the SMTP server does not support it. • `NoStartTLS` —Do not use STARTTLS; connect to the SMPT server without encryption. • `empty` —Use STARTTLS if the SMTP server supports it; otherwise, connect to the SMPT server without encryption (default value).
`grafana_smtp_enable_tracing`	No	The parameter that indicates whether to enable SMTP connection tracing for debugging email sending. Possible parameter values: • `true` —Enable SMTP connection tracing. • `false` —Disable SMTP connection tracing (default value).

Sample of the configuration file for the demonstration deployment on a single node

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
# Node on which the Kubernetes cluster and Kaspersky XDR will be deployed
nodes:
    # Node name.
  - desc: cdt-1
    # Node type.
    type: primary-worker
    # IP address of the node.
    host: 10.80.23.42
    access:
      ssh:
        # User name of the account.
        user: root
# Parameters for the Kaspersky XDR deployment
parameters:
  # IP address of the Kubernetes cluster gateway.
  - name: ingress_ip
    source:
      value: 10.80.23.182
  # IP address of the DBMS.
  - name: pg_ip
    source:
      value: 10.80.23.183
  # Path to the SSH private key for connection to the primary worker node by KDT.
  - name: ssh_pk
    source:
      path: /root/.ssh/id_rsa
  # Password of the Kaspersky XDR user account that will be created during the
  installation.
  - name: admin_password
    source:
      value: ""
  # The parameter indicating that Kaspersky XDR is installed on the target host with
  limited computing resources.
  - name: low_resources
    source:
```

```
    value: "true"
# The number of disk volumes that are used to store the service data of Kaspersky
XDR components and KDT. For the single-node deployment, set the parameter value to
"1".
- name: default_class_replica_count
  source:
    value: "1"
# The parameter indicating whether to run the secret storage in the High
Availability mode. For the single-node deployment, set the parameter value to
"false".
- name: openbao_ha_mode
  source:
    value: "false"
# The parameter indicating whether to run the secret storage in the standalone mode.
For the single-node deployment, set the parameter value to "true".
- name: openbao_standalone
  source:
    value: "true"
# Path to the KUMA inventory file that contains the installation parameters for
deployment of the KUMA services.
- name: inventory
  source:
    value: "/home/user/inventory.yml"
# Path to the license key of KUMA Core. This file is located on the administrator
host.
- name: license
  source:
    value: "/home/user/license.key"
# Domain name. The service host name and domain name are parts of FQDNs of the
Kaspersky XDR services.
- name: smp_domain
  source:
    value: "smp.local"
# List of host names of the Kaspersky XDR services for which a self-signed or custom
certificate must be generated.
- name: pki_host_list
  source:
    value: "admsrv api console kuma psql monitoring"
```

Migration to Kaspersky Next XDR Expert

This section describes the migration of data to Kaspersky Next XDR Expert from Kaspersky Security Center Windows, KUMA, and Kaspersky EDR 7.1.

About migration from Kaspersky Security Center Windows

Following this scenario, you can transfer the administration group structure, included managed devices and other group objects (policies, tasks, global tasks, tags, and device selections) from Kaspersky Security Center Windows under management of Kaspersky Next XDR Expert.

Limitations:

- Migration is only possible from Kaspersky Security Center 14.2 Windows to Kaspersky Next XDR Expert starting from version 1.0.
- You can perform this scenario only by using Kaspersky Security Center Web Console.

Stages

The migration scenario proceeds in stages:

1. **Choose a migration method**

   You migrate to Kaspersky Next XDR Expert through the Migration wizard. The Migration wizard steps depend on whether or not Administration Servers of Kaspersky Security Center Windows and Kaspersky Next XDR Expert are arranged into a hierarchy:

   - Migration by using a hierarchy of Administration Servers

     Choose this option if Administration Server of Kaspersky Security Center Windows acts as secondary to Administration Server of Kaspersky Next XDR Expert. You manage the migration process and switch between Servers within OSMP Console. If you prefer this option, you can arrange Administration Servers into a hierarchy to simplify the migration procedure. To do this, create the hierarchy before starting the migration.

     If you connect secondary Administration Servers to Kaspersky Next XDR Expert, disable background integration for Web Consoles that manage them.

     To disable background integration:

     1. Log in to Administration Server Web Console.
     2. In the main menu, go to **Settings** → **Integration**.
     3. In the **Background connection** section, disable the **Background connection for integration Enabled** toggle switch.

- Migration by using an export file (ZIP archive)

  Choose this option if Administration Servers of Kaspersky Security Center Windows and Kaspersky Next XDR Expert are not arranged into a hierarchy. You manage the migration process with two Consoles—an instance for Kaspersky Security Center Windows and OSMP Console. In this case, you will use the export file that you created and downloaded during the export from Kaspersky Security Center Windows and import this file to Kaspersky Next XDR Expert.

2. **Export data from Kaspersky Security Center Windows**

   Open Kaspersky Security Center Windows, and then run the Migration wizard.

3. **Import data to Kaspersky Next XDR Expert**

   Continue the Migration wizard to import the exported data to Kaspersky Next XDR Expert.

   If the Servers are arranged into a hierarchy, the import starts automatically after a successful export within the same wizard. If the Servers are not arranged into a hierarchy, you continue the Migration wizard after switching to Kaspersky Next XDR Expert.

4. **Perform additional actions to transfer objects and settings from Kaspersky Security Center Windows to Kaspersky Next XDR Expert manually (optional step)**

   You might also want to transfer the objects and settings that cannot be transferred through the Migration wizard. For example, you could additionally do the following:

   - Configure global tasks of Administration Server
   - Configure Network Agent policy settings
   - Create installation packages of applications
   - Create virtual Servers
   - Assign and configure distribution points
   - Configure device moving rules
   - Configure rules for auto-tagging devices
   - Create application categories

5. **Move the imported managed devices under management of Kaspersky Next XDR Expert**

   To complete the migration, move the imported managed devices under management of Kaspersky Next XDR Expert. You can do it by one of the following methods:

   - Through Kaspersky Security Center group task

     Use the Change Administration Server task to change the Administration Server to a different one for specific client devices.

   - Through the klmover utility

     Use the klmover utility and specify the connection settings for the new Administration Server.

   - Through installation or re-installation of Network Agent on the managed devices

     Create a new Network Agent installation package and specify the connection settings for the new Administration Server in the installation package properties. Use the installation package to install Network Agent on the imported managed devices through a remote installation task.

     You can also create and use a stand-alone installation package to install Network Agent locally.

6. **Update Network Agent to the latest version**

   We recommend that you upgrade the Network Agent to the same version as OSMP Console.

7. **Make sure the managed devices are visible on the new Administration Server**

   On Kaspersky Next XDR Expert Administration Server, open the managed devices list (**Asset management** → **Assets (Devices)**), and check the values in the **Visible**, **Network Agent is installed**, and **Last connected to Administration Server** columns.

Other methods of data migration

Besides the Migration wizard, you can also transfer specific tasks and policies:

- Export the task from Kaspersky Security Center Windows, and then import the tasks to Kaspersky Next XDR Expert.
- Export the policies from Kaspersky Security Center Windows, and then import the policies to Kaspersky Next XDR Expert. The related policy profiles are exported and imported together with the selected policies.

Exporting group objects from Kaspersky Security Center Windows

Migration administration group structure, included managed devices and other group objects from Kaspersky Security Center Windows to Kaspersky Next XDR Expert requires that you first select data for exporting and create an export file. The export file contains information about all group objects that you want to migrate. The export file will be used for subsequent import to Kaspersky Next XDR Expert.

You can export the following objects:

- Tasks and policies of managed applications
- Global tasks
- Custom device selections
- Administration group structure and included devices
- Tags that have been assigned to migrating devices

Before you start exporting, read general information about migration to Kaspersky Next XDR Expert. Choose the migration method—by using or not using the hierarchy of Administration Servers of Kaspersky Security Center Windows and Kaspersky Next XDR Expert.

To export managed devices and related group objects through the Migration wizard:

1. Depending on whether or not the Administration Servers of Kaspersky Security Center Windows and Kaspersky Next XDR Expert are arranged into a hierarchy, do one of the following:

   - If the Servers are arranged into a hierarchy, open OSMP Console, and then switch to the Server of Kaspersky Security Center Windows.

   - If the Servers are not arranged into a hierarchy, open Kaspersky Security Center Web Console connected to Kaspersky Security Center Windows.

2. In the main menu, go to **Operations** → **Migration**.

3. Select **Migrate to Kaspersky Security Center Linux or Open Single Management Platform** to start the wizard and follow its steps.

4. Select the administration group or subgroup to export. Please make sure that the selected administration group or subgroup contains no more than 10,000 devices.

5. Select the managed applications whose tasks and policies will be exported. Select only applications that are supported by Kaspersky Next XDR Expert. The objects of unsupported applications will still be exported, but they will not be operable.

6. Use the links on the left to select the global tasks, device selections, and reports to export. The **Group objects** link allows you to exclude custom roles, internal users and security groups, and custom application categories from the export.

The export file (ZIP archive) is created. Depending on whether or not you perform migration with Administration Server hierarchy support, the export file is saved as follows:

- If the Servers are arranged into a hierarchy, the export file is saved to the temporary folder on OSMP Console Server.

- If the Servers are not arranged into a hierarchy, the export file is downloaded to your device.

For migration with Administration Server hierarchy support, the import starts automatically after a successful export. For migration without Administration Server hierarchy support, you can import the saved export file to Kaspersky Next XDR Expert manually.

Importing the export file to Kaspersky Next XDR Expert

To transfer information about managed devices, objects, and their settings that you exported from Kaspersky Security Center Windows, you must import it to Kaspersky Next XDR Expert.

To import managed devices and related group objects through the Migration wizard:

1. Depending on whether or not the Administration Servers of Kaspersky Security Center Windows and Kaspersky Next XDR Expert are arranged into a hierarchy, do one of the following:

   - If the Servers are arranged into a hierarchy, proceed to the next step of the Migration wizard after the export is completed. The import starts automatically after a successful export within this wizard (see step 2 of this instruction).

   - If the Servers are not arranged into a hierarchy:

     a. Open OSMP Console.

     b. In the main menu, go to **Operations** → **Migration**.

     c. Select the export file (ZIP archive) that you created and downloaded during the export from Kaspersky Security Center Windows. The upload of the export file starts.

2. After the export file is uploaded successfully, you can continue importing. If the Servers are not arranged into a hierarchy, you can specify another export file by clicking the **Change** link, and then selecting the required file.

3. The entire hierarchy of administration groups of Kaspersky Next XDR Expert is displayed.

   Select the check box next to the target administration group to which the objects of the exported administration group (managed devices, policies, tasks, and other group objects) must be restored.

4. The import of group objects starts. You cannot minimize the Migration wizard and perform any concurrent operations during the import. Wait until the refresh icons (🔄) next to all items in the list of objects are replaced with green check marks (✓) and the import finishes.

5. When the import completes, the exported structure of administration groups, including device details, appears under the target administration group that you selected. If the name of the object that you restore is identical to the name of an existing object, the restored object has an incremental suffix added.

   If in a migrated task the details of the account under which the task is run are specified, you have to open the task and enter the password again after the import is completed.

If the import has completed with an error, you can do one of the following:

- For migration with Administration Server hierarchy support, you can start to import the export file again. In this case, you have to select the administration group as described at step 3.

- For migration without Administration Server hierarchy support, you can start the Migration wizard to select another export file, and then import it again.

You can check whether the group objects included in the export scope have been successfully imported to Kaspersky Next XDR Expert. To do this, go to the **Asset management** section and ensure whether the imported objects appear in the corresponding subsections.

Note that the imported managed devices are displayed in the **Managed devices** subsection, but they are invisible in the network and Network Agent is not installed and running on them (the *No* value in the **Visible**, **Network Agent is installed**, and **Last connected to Administration Server** columns).

To complete the migration, you need to switch the managed devices to be under management of Kaspersky Next XDR Expert as described at stage 5 in Migration to Kaspersky Next XDR Expert.

Switching managed devices to be under management of Kaspersky Next XDR Expert

After a successful import of information about managed devices, objects, and their settings to Kaspersky Next XDR Expert, you need to switch the managed devices to be under management of Kaspersky Next XDR Expert to complete the migration.

You can move the managed devices to be under Kaspersky Next XDR Expert by one of the following methods:

- Using the klmover utility.
- Using the *Change Administration Server* task.
- Installing Network Agent on the managed devices through a remote installation task.

To switch managed devices to be under management of Kaspersky Next XDR Expert by installing Network Agent:

1. Remove Network Agent on the imported managed devices that will be switched under management of Kaspersky Next XDR Expert.
2. Switch to Administration Server of Kaspersky Security Center Windows.
3. Go to **Discovery & deployment** → **Deployment & assignment** → **Installation packages**, and then open the properties of an existing installation package of Network Agent.

   If the installation package of Network Agent is absent in the package list, download a new one.

   You can also create and use a stand-alone installation package to install Network Agent locally.
4. On the **Settings** tab, select the **Connection** section. Specify the connection settings of Administration Server of Kaspersky Next XDR Expert.
5. Create a remote installation task for imported managed devices, and then specify the reconfigured Network Agent installation package.

   You can install Network Agent through Administration Server of Kaspersky Security Center Windows or through a Windows-based device that acts as a distribution point. If you use Administration Server, enable the **Using operating system resources through Administration Server** option. If you use a distribution point, enable the **Using operating system resources through distribution points** option.
6. Run the remote installation task.

After the remote installation task finishes successfully, go to Administration Server of Kaspersky Next XDR Expert and ensure that managed devices are visible in the network, and that Network Agent is installed and running on them (the *Yes* value in the **Visible**, **Network Agent is installed**, and **Network Agent is running** columns).

About migration from KEDR 7.1

This section covers the migration process from KEDR 7.1 to Kaspersky Next XDR Expert 2.0. It includes data migration scenarios for OSMP deployed on a single node and as a distributed solution, as well as guidance on telemetry migration.

Before upgrading from KEDR 7.1 to Kaspersky Next XDR Expert 2.0, you must first update the Kaspersky Endpoint Security Endpoint Agents. Only the latest versions of Endpoint Agents are capable of sending the updated telemetry data required by Kaspersky Next XDR Expert 2.0. Earlier versions are not compatible with this release.

We recommend migrating only alerts with a **Closed** status.

Migration implies three stages:

1. Preparation of an archive to export from KEDR 7.1.
2. Importing data to Kaspersky Next XDR Expert 2.0.
3. Telemetry data migration (optional stage): using an external storage or without external storage.

Transferred data

During the upgrade and migration from KEDR 7.1 to Kaspersky Next XDR Expert 2.0, the following data is transferred:

- Alert data:
  - Alerts related to files received from endpoint devices (including KSB scenarios) and from external systems via API
  - IOC alerts

    IOC alerts related to syntactically incorrect IOC rules (that do not fully comply with the OpenIOC version 1.0 and 1.1 standards) are not migrated. Information about such IOC alerts is displayed in the error log.
  - IOA alerts

  We recommend migrating only alerts with a **Closed** status.
- Additional alert data:
  - Sandbox storage
  - Detected file storage

- Information about IOC rules:
  - Name
  - Description
  - Severity
  - IOC rule
  - Scheduled scanning option (enabled or disabled)
  - Scope information for global IOC rules (within the tenant or across selected Secondary Central Nodes)

  IOC rules are transferred only if they fully comply with the OpenIOC version 1.0 and 1.1 standards. Alerts associated with non-compliant IOC alerts are not transferred.

- Information about custom IOA rules:
  - Name
  - Description
  - Severity
  - Confidence
  - Trigger
  - Indication of whether markup is enabled by this rule
  - Scope information for global IOA rules (within the tenant or across selected Secondary Central Nodes)
- Information about IOA exclusions:
  - ID of the IOA rule to which the exclusion applies
  - Exclusion type (unconditional or conditional)
  - Exclusion condition
  - If both global and local exclusions exist for a rule, only the global exclusion is exported
  - Whether the exclusion is enabled (applies to conditional exclusions only; disabled unconditional exclusions are not exported)

  > IOA exclusions can be imported as either common exclusions or event-type-specific exclusions. If you cannot find an imported exclusion in one section, check the other section.

- Information about custom YARA rules:
  - Name
  - Description

    - Content
    - Whether the rule runs in stream mode
    - Scope information for global YARA rules (within the tenant or across selected Secondary Central Nodes)
- Information about custom Sandbox submission rules (scanning on custom images):
    - For files:
        - Whether the rule is enabled
        - Custom image name
        - File mask for submission
        - Exclusion mask
        - File category
        - File size ranges
    - For links:
        - Custom image name
        - Whether submission to this image is enabled
- List of image names included in the default Sandbox set
- Repository containing forensic artifacts, manually uploaded files, and files obtained via the `get` task:
    - File name
    - File size
    - MD5 hash
    - SHA-256 hash
    - File date
    - Name of the user who uploaded the file
    - Scan results from Anti-Malware Engine, YARA, and Sandbox
- Information about prevention rules:
    - Status (enabled or disabled)
    - Name
    - Hash

    - Whether notifications are displayed on devices
    - Scope (devices, servers, tenants)
- Information about archive passwords used for anti-malware and Sandbox scanning
- Information about active system users (excluding AD users), including:
    - User name
    - User role
    - User type (internal or Active Directory)
    - List of tenants the user has access to
- Setting that enables or disables file processing within KSB scenarios. Setting options:
    - File processing disabled
    - High-priority file processing enabled (manual submission scenario)
    - File processing enabled for all priorities (all scenarios)
- IOC file scanning schedule
    - Start time
    - Scan duration
- Email notification templates
    - Template status (enabled or disabled)
    - Recipients
    - Subject
    - Notification type (component health or detections)
    - Delivery criteria
        - For component health:
            - All
            - Selected components: AV, YARA, Sandbox, KSN, TAA, disk usage, CPU usage, RAM usage, database updates, Sensor servers, cluster servers, queues, authorization policies
        - For detections:
            - All
            - Selected components: AV, YARA, Sandbox, IOC, TAA

Rules that have no supported criteria selected, and only criteria not supported by OSMP (for example, IDS), will not be exported.

- For detections only:
  - Detection severity
  - Source or destination address (mask)
  - Sender or recipient address
- Information about Endpoint Agents:
  - Agent name (FQDN)
  - IP address
  - Operating system name
  - Sensor ID
  - Tenant or Secondary Central Node where the agent is registered (in distributed deployments)
- Information about Endpoint Agent groups and tags:
  - Information about groups:
    - Name
    - Members
      A group may include:
      - Endpoint Agents
      - Other assets
      - Nested groups (nesting depth is not limited)
  - Information about tags:
    - Name
    - List of Endpoint Agents assigned this tag

Information about assets other than Endpoint Agents that are part of groups or have tags is not transferred.

Information about groups that do not contain any Endpoint Agents at any level of nesting is not transferred.

- Threat Hunting telemetry database
  - For events related to IOA detections, only the **Severity** and **Confidence** of the IOA rule are shown.

Please note, the Quarantine is not transferred.

Hardware requirements for data migration

This section covers the hardware and disk space requirements necessary for successful data migration.

This calculation is given for the single-node deployment scheme.

External disk requirements

- **Size:**

  You must allocate additional disk space to temporarily store data during the migration process. To calculate the minimum required disk size, use the following formula:

  `Disk_size_GB = 2 * ( (N_ioa_alerts + N_ioc_alerts) * 0.00005GB + N_ext_alerts * 0.2GB + 0.8 * Storage_GB + 1GB)`

  Where:

  - `Disk_size_GB` is the minimum disk size.
  - `N_ioa_alerts` is the number of alerts in the `ioa_alerts` table.
  - `N_ioc_alerts` the number of alerts in the `ioc_scan_ep_alert` table.
  - `N_ext_alerts` is the number of alerts obtained via External API detections.
  - `Storage_GB` is the size of the object storage (S3) specified in the KATA web interface. Of this, 40% is occupied by `edr_storage_bucket`, and 20% each is used by `am-quarantine-bucket` and `sb-artifacts-bucket`.
  - `1GB` is the maximum total size for other exported asset data, settings, and rules.
  - Multiplication by two at the beginning of the formula accounts for the additional space required to create the export archive.

- **Hardware:**

  Hardware requirements vary depending on the telemetry migration scenario. The following requirements apply to the scenario in which telemetry is transferred using an external disk.

  - Type: SSD (RAID 0).
  - Performance: 200 ROPS (read operations per second), 800 WOPS (write operations per second), ~3GB/s.
  - Required disk size is calculated by multiplying the Elasticsearch data volume by 2.6. The total data volume in Elasticsearch can be calculated using the following formula:

    `<effective number of hosts>/15000 * (460GB * <data storage period, in days>)/0.65`

Where:

- `<effective number of hosts>` is determined by your IT infrastructure.
- `<data storage period, in days>` refers to the `telemetry-keep-days` parameter specified during the export.

Requirements for the import

To perform the data migration, the resources listed below must be reserved on the Kaspersky Next XDR Expert 2.0 server to which you will transfer the data.

> The following calculations do not include resources required for telemetry data migration. We recommend allocating double the storage capacity calculated based on the sizing guide.

- CPU: +4 cores.
- RAM: +4 GB.
- Available disk space: `Storage_GB`, the size of the object storage (S3) specified in the KATA web interface.

> Reservation implies that either the current Kaspersky Next XDR Expert 2.0 load allows the specified resources to be available, or the import should be performed during a period of reduced load (outside working hours). Also, the resources required for import can be ensured by running data import during periods of low server load.

Requirements for telemetry migration

The telemetry migration scenario that involves creating a new KUMA Collector (elastic type) requires additional resources.

A separate connector must be created for each index in Elasticsearch. As there can be many indexes, we recommend using the script to export and import telemetry using a separate external storage (specify the `--step third` parameter when running the script).

To obtain a list of indexes, run the following command:

```
docker exec -t $(docker ps -q -f name=elasticsearch.1) curl localhost:9200/_cat/indices?h=index,store.size
```

You can create and use multiple connectors to import data. We recommend using up to five connectors simultaneously.

To ensure the stability of the telemetry migration process, the following requirements must be met on the Kaspersky Next XDR Expert 2.0 server to which you will transfer the data:

- For the scenario of transferring telemetry without external storage:
  - When exporting from one index, the transfer speed is 9,000 EPS and the following additional resources are required:
    - CPU: +6 cores.
    - RAM: +4 GB.
  - When exporting from five indexes simultaneously, the transfer speed is 20,000 EPS and the following additional resources are required:
    - CPU: +12 cores.
    - RAM: +8 GB.

  The additional CPU and RAM requirements can be reduced by moving the Elasticsearch collector to a separate server and adding this server to the KUMA cluster according to the instructions. In this case, the following resources are required:

  - Resources for a separate host:
    - for one index: 5 cores, 1GB RAM.
    - for five indexes: 10 cores, 1GB RAM.
  - Kaspersky Next XDR Expert 2.0 server sizing increase:
    - for one index: +2 cores, +4GB RAM.
    - for five indexes: +4 cores, +8GB RAM.

  Once the migration is finished, the connector can be deleted and the server can be removed from the KUMA cluster.

- For the scenario of transferring telemetry using the external storage, the transfer speed is 40,000 EPS and the following sizing increase is required:
  - Kaspersky Next XDR Expert 2.0 server sizing increase:
    - CPU: +18 cores.
    - RAM: +12 GB.
  - With KUMA Collector on a separate host:
    - Resources for a separate host: 12 cores, 1GB RAM.
    - Kaspersky Next XDR Expert 2.0 server sizing increase: +6 cores, +12GB RAM.

- When transferring telemetry data using an external storage, KUMA Collector resources can be limited, regardless of whether the collector is on a separate host or on the OSMP server. We recommend limiting the number of processor cores used by the KUMA Collector to two CPUs. Note that this reduces the data transfer speed by approximately 2.7 times to 15,000 EPS.

  - For KUMA Collector on the OSMP server, increase the OSMP server size by:

    - CPU: +4 cores.

    - RAM: +8 GB.

  - With KUMA Collector on a separate host:

    - Resources for a separate host: 3 cores, 1GB RAM.

    - OSMP server sizing increase: +3 cores, +8GB RAM.

  To limit the number of CPUs, run the following command:

  ```
  systemctl set-property kuma-collector-<id>.service CPUQuota=<number of cores>*100%
  ```

  For example, to limit CPU by 200% (equal to 2 cores), run the following command:

  ```
  systemctl set-property kuma-collector-<id>.service CPUQuota=200%
  ```

- Network bandwidth: at least 800 Mbps.

- Events from Elasticsearch are stored in ClickHouse at approximately the same size as events received through a standard collector. The size ratio is 1:1.

- Additional telemetry storage:

  - The storage logic for data migrated from KATA depends on the configuration of the KUMA Storage spaces (including retention) where the KUMA Collector (elastic type) will store the data.

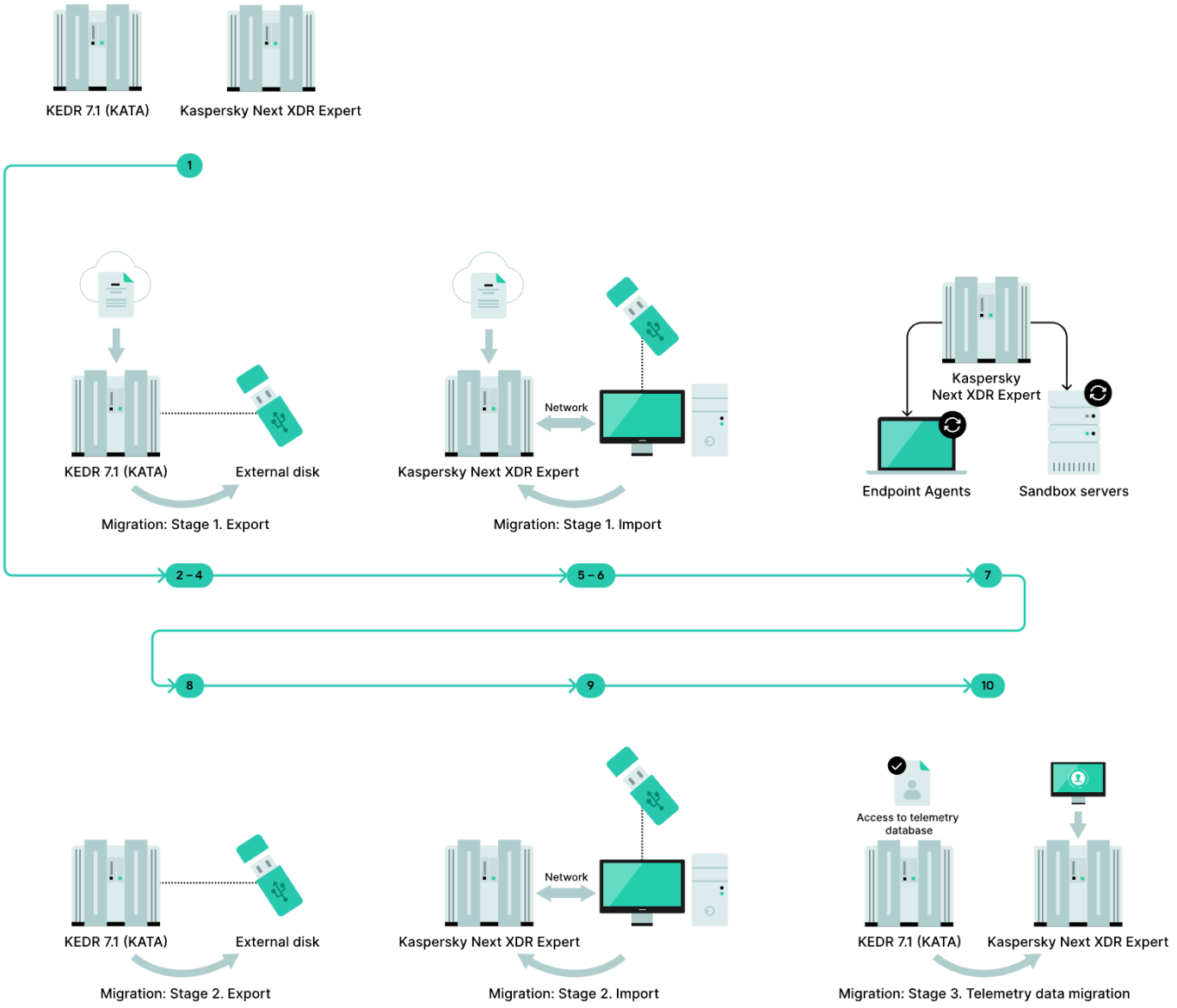
  - The transferred telemetry must be located in a separate space.

  - We recommend allocating double the storage capacity based on the sizing guide due to the specifics of the data transfer process (as sequential data transfer is not guaranteed).

Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using a separate server for Kaspersky Next XDR Expert 2.0

This section covers the data migration scenario from KEDR 7.1 operating on the KATA platform to Kaspersky Next XDR Expert 2.0 running on OSMP, using a separate server for OSMP. Use this scenario if your infrastructure has sufficient hardware resources and the main priority is to minimize Kaspersky Next XDR Expert functionality downtime.

The stages of the migration process are shown in the picture below.



Migration stages

Migrating data using a separate server for OSMP minimizes application downtime during the upgrade. To achieve downtime minimization, the migration process is divided into three phases:

1. Migrating the data necessary to run the Kaspersky Next XDR Expert 2.0 and ensure the same protection level as the upgraded KEDR 7.1.

2. Migrating historical data.

3. Migrating telemetry data.

Therefore, the general migration flow is as follows:

1. Saving the data necessary for the application to operate.

2. Importing the saved data into OSMP.

   While steps 1 and 2 can be completed relatively quickly, protection is still provided during this time by KEDR 7.1.

3. Switching Endpoint Agents via Kaspersky Security Center policies to the new server.

   The period during which Endpoint Agents are switched is considered downtime. This process is described in stage 8 of the scenario below.

4. Stages 8-10 in the scenario involve migrating historical data and telemetry. While these stages can take a significant amount of time, by this phase, OSMP is already fully operational and functional.

Limitations

Data migration to OSMP deployed on a single node scenario has the following limitations:

- Impossible to release the KEDR 7.1 server until the data migration process is finished.
- The following actions should be avoided until the export process is finished:
  - Add, edit or delete users
  - Perform any actions with operating system images in the Sandbox or modify Sandbox rules
  - Add, edit or delete prevention rules
  - Add, edit or delete IOC, IOA rules, YARA rules, IOA exclusions, and IOC scan schedule
  - Add or remove Endpoint Agents, or edit Endpoint Agent groups and tags
  - Change the list of archive passwords

Migration stages

Data migration to OSMP deployed on a single node proceeds in stages:

1. **OSMP deployment on a single node**

   This scenario only covers the single-node OSMP deployment. Data migration for multitenancy mode is described in a separate scenario.

   Deploy Kaspersky Next XDR Expert.

2. **Mounting an external disk to the KEDR 7.1 server**

   If you prefer to use a shared folder as an external storage, create it on a separate Linux server.

   How to create a network share

   To create a Samba share:

   1. Install Samba:

      `sudo apt update`

      `sudo apt install samba samba-common-bin`

   2. Add a Samba user with creating home folder for this user. This folder will be the shared folder.

      Please do not create a shared folder manually as the current user. You must do this using the command below:

      `sudo useradd -m -d /home/<SHARE_FOLDER> -s /bin/bash <USER_NAME>`

3. Set a password and activate a user:

   `sudo smbpasswd -a <USER_NAME>`

   `sudo smbpasswd -e <USER_NAME>`

4. Add the following block to the configuration file /etc/samba/smb.conf:

   `[<SHARE_FOLDER>]`

   `comment = Network share`

   `path = /home/<SHARE_FOLDER>`

   `read only = no`

   `browseable = yes`

   `guest ok = no`

   where:

   - `SHARE_FOLDER` is the name of a shared folder.

5. Restart the service:

   `sudo systemctl restart smbd nmbd`

To create a Windows Share (cmd/PowerShell):

1. Make sure the console is running as administrator.

2. Create a user and set a password:

   `net user <USER_NAME> <PASSWORD> /add /expires:never`

3. Create a folder and share it:

   `mkdir <SHARE_FOLDER>`

   `icacls <SHARE_FOLDER> /inheritance:r`

   `icacls <SHARE_FOLDER> /grant "Everyone:(OI)(CI)F"`

4. Create a shared resource for a user:

   `net share <SHARE_NAME>=<SHARE_FOLDER> /grant:<USER_NAME>,full`

In Technical Support Mode, mount an external disk (physical or network share) to the server where KEDR 7.1 is deployed.

How to mount and unmount a physical disk

To mount a physical disk:

1. Connect a physical hard drive to the host and find out its system name.

2. Search for the disk by its size in the command output:

   `lsblk`

   `sudo fdisk -l`

   Example of a found device: /dev/sdb1

3. Format the device to the Ext4 file system to prepare the disk for writing data:

   `sudo mkfs.ext4 /dev/<disk>`

4. To create a mount point, create an empty directory where the disk will be mounted:

   `sudo mkdir -p /mnt/external_disk`

5. Run the following command depending on the disk file system:

   `sudo mount /dev/<disk> /mnt/external_disk`

To unmount a physical disk after performing required operations, run the following command:

`sudo umount /mnt/external_disk`

> We strongly recommend to run this command before physically disconnect the cable to avoid data loss.

How to mount and unmount a network share

This method is intended for SMB/CIFS (Windows Share/Samba). It is suitable for a one-time connection. The connection will be lost after a system reboot.

To mount a network share:

1. Download and install the following three packages:

   `libtalloc2_2.4.2-1~bpo12+1_amd64.deb`

   `libwbclient0_4.9.5+dfsg-5+deb10u3astra.ce4_amd64.deb`

   `cifs-utils_6.8-2+deb10u1+ci202212062128+astra2_amd64.deb`

   > Do not modify the source.list file. To install software, download the required .deb packages from public repositories or from private repositories supported by your organization.

2. Create a mount point:

   `sudo mkdir -p <PATH_FOLDER>`

3. Mount a network share:

   `sudo mount -t cifs //<SERVER>/<SHARE_FOLDER> /<PATH_FOLDER> -o username=<NAME>,password=<PASSWORD>,vers=2.1`

   where:

   - `SERVER` is the server's IP address or its network name.
   - `SHARE_FOLDER` is the name of a shared folder.
   - `PATH_FOLDER` is the local path to the mount point.
   - `NAME` is the user name on Windows/Samba.
   - `PASSWORD` is the user password.

- `2.1` is the recommended protocol version.

  Command example:

  ```
  sudo mount -t cifs //<IP address>/Share /mnt/share -o username=<user name>,password=<password>,vers=2.1
  ```

To unmount a network share, run the following command:

```
sudo umount <PATH_FOLDER>
```

After completing the migration, it is recommended to remove the cifs-utils package and its dependencies.

To delete the cifs-utils package and its dependencies, run the following command:

```
sudo apt remove --purge cifs-utils && sudo apt autoremove -y
```

3. **Placing scripts and files required for data migration on the KATA/EDR server**

In Technical Support Mode, place the export script and TAR archive containing Docker image img_kata_osmp_migrate.tar on the KATA server.

The data export script performs a two-step export:

1. Exporting data required to start OSMP and replicate the KEDR 7.1 operational state:
   - List of active (enabled) users
   - List of image names included in the Sandbox reference set
   - Sandbox custom rules (scanning on custom images)
   - Setting that enables or disables file scanning in the Sandbox
   - Prevention rules
   - IOC rules
   - Custom IOA rules
   - Custom YARA rules
   - Passwords for archives used during anti-virus and Sandbox scan
   - IOC scan schedule
   - List of Endpoint Agents
   - Endpoint Agent groups and tags (assets)
2. Historical data and additional settings:
   - Alert database
   - Additional data related to alerts

- Repository (forensics, manually downloaded files, and files previously obtained via the `get` task)
- Email notification templates

4. **Running the export script for the first step**

   Run the export script for the first step (specify the `--step first` parameter when running the kata-osmp-export.sh script), and provide the address of the mounted disk as the target location for the exported data.

   The script exports IOA exclusions (excludes.json) to a separate file in a format compatible with Endpoint Agent policy, allowing them to be imported later. This file can be found in the `ioa` folder of the exported data.

5. **Linux server deployment**

   Set up a separate Linux server.

   Configure the integration between Linux the server and OSMP.

   Install Docker on the Linux server.

   Place the import script and a TAR archive containing Docker image img_kata_osmp_migration_importer.tar on the separate Linux server.

   If KEDR 7.1 has not yet been replaced by Kaspersky Next XDR Expert 2.0 and is still running, it can be used as the required Linux server. In this case there is no need to install Docker, as it is already installed on the KEDR 7.1.

6. **Running the import script for the first step**

   If you are using separate Linux server, unmount the external disk from KEDR 7.1 in Technical Support Mode and mount it to the Linux server.

   On the Linux server, run the import script for the first step.

7. **Preparing for the second step of data export**

   After successful completion of the first data import step, perform the following actions:

   - Disconnect KEDR 7.1 from all external integrations.
   - Disconnect Endpoint Agents in policy settings by deleting Endpoint Agent certificates and enabling certificate authentication. After this, Endpoint Agents will not be able to connect to the server.
   - If your Endpoint Agents versions are lower than those compatible with Kaspersky Next XDR Expert 2.0, you need to update them.
   - **Import custom IOA exclusions via Endpoint Agents policies**.

     1. Open the archive containing data to migrate.
     2. Extract the ioa.json file containing exclusions.
     3. Import the extracted file to KES policies (KES for Windows, KES for Linux).

   - Connect Endpoint Agents to OSMP server via Kaspersky Security Center server.

8. **Running the export script for the second step**

   If you are using separate Linux server, unmount the external disk from this server and in Technical Support Mode mount it to the KEDR 7.1 server.

   Run the export script for the second step (specify the `--step second` parameter when running the kata-osmp-export.sh script).

9. **Running the import script for the second step**

   If you are using separate Linux server, unmount the external disk from KEDR 7.1 in Technical Support Mode and mount it to the Linux server.

   On the Linux server, run the import script for the second step.

10. **Migrating telemetry data**

    You may migrate telemetry data by using external storage or without external storage.

After successful completion of all export and import operations and telemetry transfer, you can reuse the KEDR 7.1 server.

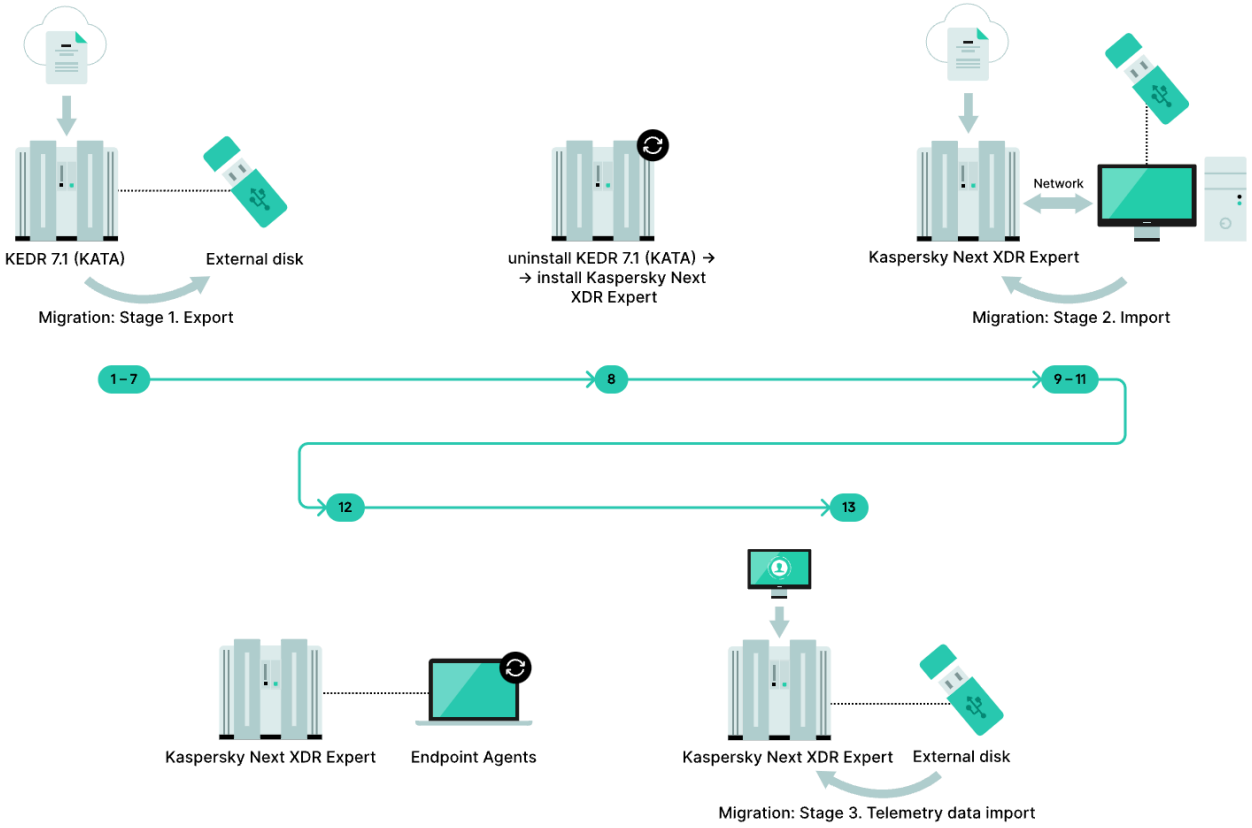
> If you have KATA or KATA/NDR licenses, you can upgrade KATA/KEDR 7.1 to KATA 8.0. In this case, KEDR functionality will no longer be available: only KATA data will be transferred.

Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using the same server for Kaspersky Next XDR Expert 2.0

This section covers the data migration scenario from KEDR 7.1 operating on the KATA platform to Kaspersky Next XDR Expert 2.0 running on OSMP, using the same server for OSMP/Kaspersky Next XDR Expert 2.0.

This scenario allows for hardware reuse, but requires significant server downtime. In this scenario, OSMP is deployed directly on the existing Kaspersky Next XDR Expert server after the data export stage.

The stages of migration are shown in the picture below.



Migration stages

Limitations

Data migration to OSMP deployed on Kaspersky Next XDR Expert server scenario has the following limitations:

- Data can be transferred both with or without telemetry. If you want to migrate telemetry data, perform the third export step (use the `--step third` parameter) and then use the separate scenario.
- Estimated Kaspersky Next XDR Expert downtime is up to six hours without telemetry transfer. With telemetry transfer, downtime depends on the size of accumulated telemetry and takes approximately 24 hours for a configuration with 5000 Endpoint Agents and a telemetry storage depth of 30 days.
- Additional Linux host is required to start the data import into OSMP.
- We do not recommend performing any data modifications until the import process is finished.

Stages

Data migration to OSMP deployed on the Kaspersky Next XDR Expert server involves the following stages:

1. **Mounting an external disk to the KEDR 7.1 server**

   If you prefer to use a shared folder as an external disk, create it on a separate Linux server.

   How to create a network share

To create a Samba share:

1. Install Samba:

   `sudo apt update`

   `sudo apt install samba samba-common-bin`

2. Add a Samba user with creating home folder for this user. This folder will be the shared folder.

   Please do not create a shared folder manually as the current user. You must do this using the command below:

   `sudo useradd -m -d /home/<SHARE_FOLDER> -s /bin/bash <USER_NAME>`

3. Set a password and activate a user:

   `sudo smbpasswd -a <USER_NAME>`

   `sudo smbpasswd -e <USER_NAME>`

4. Add the following block to the configuration file /etc/samba/smb.conf:

   `[<SHARE_FOLDER>]`

   `comment = Network share`

   `path = /home/<SHARE_FOLDER>`

   `read only = no`

   `browseable = yes`

   `guest ok = no`

   where:

   - `SHARE_FOLDER` is the name of a shared folder.

5. Restart the service:

   `sudo systemctl restart smbd nmbd`

To create a Windows Share (cmd/PowerShell):

1. Make sure the console is running as administrator.

2. Create a user and set a password:

   `net user <USER_NAME> <PASSWORD> /add /expires:never`

3. Create a folder and share it:

   `mkdir <SHARE_FOLDER>`

   `icacls <SHARE_FOLDER> /inheritance:r`

   `icacls <SHARE_FOLDER> /grant "Everyone:(OI)(CI)F"`

4. Create a shared resource for a user:

   `net share <SHARE_NAME>=<SHARE_FOLDER> /grant:<USER_NAME>,full`

In Technical Support Mode, mount an external disk (physical or network share) to the server where KEDR 7.1 is deployed.

How to mount and unmount a physical disk

To mount a physical disk:

1. Connect a physical hard drive to the host and find out its system name.
2. Search for the disk by its size in the command output:

   `lsblk`

   `sudo fdisk -l`

   Example of a found device: /dev/sdb1
3. Format the device to the Ext4 file system to prepare the disk for writing data:

   `sudo mkfs.ext4 /dev/<disk>`
4. To create a mount point, create an empty directory where the disk will be mounted:

   `sudo mkdir -p /mnt/external_disk`
5. Run the following command depending on the disk file system:

   `sudo mount /dev/<disk> /mnt/external_disk`

To unmount a physical disk after performing required operations, run the following command:

`sudo umount /mnt/external_disk`

> We strongly recommend to run this command before physically disconnect the cable to avoid data loss.

How to mount and unmount a network share

This method is intended for SMB/CIFS (Windows Share/Samba). It is suitable for a one-time connection. The connection will be lost after a system reboot.

To mount a network share:

1. Download and install the following three packages:

   `libtalloc2_2.4.2-1~bpo12+1_amd64.deb`

   `libwbclient0_4.9.5+dfsg-5+deb10u3astra.ce4_amd64.deb`

   `cifs-utils_6.8-2+deb10u1+ci202212062128+astra2_amd64.deb`

   > Do not modify the source.list file. To install software, download the required .deb packages from public repositories or from private repositories supported by your organization.
2. Create a mount point:

   `sudo mkdir -p <PATH_FOLDER>`
3. Mount a network share:

   `sudo mount -t cifs //<SERVER>/<SHARE_FOLDER> /<PATH_FOLDER> -o username=<NAME>,password=<PASSWORD>,vers=2.1`

where:

- `SERVER` is the server's IP address or its network name.
- `SHARE_FOLDER` is the name of a shared folder.
- `PATH_FOLDER` is the local path to the mount point.
- `NAME` is the user name on Windows/Samba.
- `PASSWORD` is the user password.
- `2.1` is the recommended protocol version.

  Command example:

  ```
  sudo mount -t cifs //<IP address>/Share /mnt/share -o username=<user name>,password=<password>,vers=2.1
  ```

To unmount a network share, run the following command:

```
sudo umount <PATH_FOLDER>
```

After completing the migration, it is recommended to remove the cifs-utils package and its dependencies.

To delete the cifs-utils package and its dependencies, run the following command:

```
sudo apt remove --purge cifs-utils && sudo apt autoremove -y
```

2. **Placing scripts and files required for data migration on the KATA/KEDR server**

In Technical Support Mode, place the export script and TAR archive containing Docker image img_kata_osmp_migrate.tar on the KATA server.

The data export script performs a three-step export:

1. Exporting data required to start OSMP and replicate the KEDR 7.1 operational state:
   - List of active (enabled) users
   - List of image names included in the Sandbox reference set
   - Sandbox custom rules (scanning on custom images)
   - Setting that enables or disables file scanning in the Sandbox
   - Prevention rules
   - IOC rules
   - Custom IOA rules
   - Custom YARA rules
   - Passwords for archives used during anti-virus and Sandbox scan

- IOC scan schedule
- List of Endpoint Agents
- Endpoint Agent groups and tags (assets)

2. Historical data and additional settings:
   - Alert database
   - Additional data related to alerts
   - Repository (forensics, manually downloaded files, and files previously obtained via the `get` task)
   - Email notification templates
3. Telemetry data

3. **Running the export script for the first step**

   Run the export script for the first step (specify the `--step first` parameter when running the kata-osmp-export.sh script), specifying the address of the mounted disk as the target location for the exported data.

   The script exports IOA exclusions (excludes.json) to a separate file in a format compatible with Endpoint Agent policy, allowing them to be imported later. This file can be found in the `ioa` folder of the exported data.

   The script provides a notification upon successful completion of the first data export step.

4. **Preparing for the second data export step**

   After successful completion of the first data export step, perform the following actions:

   - Disconnect KEDR 7.1 from all external integrations.
   - Disconnect Endpoint Agents in policy settings by deleting Endpoint Agent certificates and enabling certificate authentication. After this, Endpoint Agents will not be able to connect to the server.

5. **Running the export script for the second step**

   Run the export script for the second step (specify the `--step second` parameter when running the kata-osmp-export.sh script).

6. **Running the export script for the third step**

   If you want to transfer telemetry data, run the export script for the third step (specify the `--step third` parameter when running the kata-osmp-export.sh script).

   The telemetry data will be under the path migration-dir/nodes/[GUID of the node]/archive_source/elasticsearch/ in the .dump files.

7. **Unmounting an external disk from the KEDR 7.1 server**

   In Technical Support Mode, unmount the external disk from the KEDR 7.1 server.

8. **OSMP deployment on the KATA/EDR server**

   Uninstall KEDR 7.1 by removing it from the disk, deploy OSMP on the released KATA/KEDR server, and update Sandbox servers.

   It is assumed that there is only one tenant in the system.

9. **Linux server deployment**

Set up a separate Linux server.

Configure the integration between Linux server and OSMP.

Install Docker on the Linux server.

Place the import script and TAR archive containing Docker image img_kata_osmp_migration_importer.tar on the separate Linux server.

10. **Mounting an external disk to the Linux server**

    Mount the external disk to the Linux server to import data.

11. **Running the import script**

    On the Linux server, run the import script.

12. **Performing steps after import**

    Perform the following actions:

    - If your Endpoint Agents versions are lower than those compatible with Kaspersky Next XDR Expert 2.0, you need to update them.
    - **Import custom IOA exclusions via Endpoint Agents policies**.

      1. Open the archive containing data to migrate.
      2. Extract the ioa.json file containing exclusions.
      3. Import the extracted file to KES policies (KES for Windows, KES for Linux).

    - Connect Endpoint Agents to OSMP server via Kaspersky Security Center server.

13. **Migrating telemetry data**

    To import telemetry data exported on the previous step, use the separate scenario.

If you have KATA or KATA/NDR licenses, you can upgrade KATA/KEDR 7.1 to KATA 8.0. In this case, Kaspersky Next XDR Expert functionality will no longer be available: only KATA data will be transferred.

Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using the same server for Kaspersky Next XDR Expert 2.0 in a distributed solution

This section covers the data migration scenario for Open Single Management Platform deployed in a distributed solution. Unlike single-node scenarios, distributed deployments involve multiple nodes (PCN, SCN, Embedded SCN) and require careful tenant mapping to preserve data isolation and hierarchy.

Limitations

Data migration in a distributed solution mode scenario has the following limitations:

- For all data migration except telemetry:
  - The same external disk (physical or network storage) must be used to transfer distributed solution data.
  - To export data, the export script must be run sequentially on each node of the distributed solution. The sequence of nodes is arbitrary and can be chosen for operational convenience.
  - Data is imported using an external disk containing all exported data from all nodes.
  - Data import must be initiated via the OSMP node that contains the root tenant. Data belonging to secondary nodes and tenants is also imported via the root node and propagated down the tenant hierarchy.
  - When YARA rules are used for traffic scanning on a multitenant OSMP server, they are migrated to the server's Root tenant, even if they were created on an SCN linked to a second-level tenant.
  - When archive passwords are used for file scanning in a feed on a multitenant OSMP server, they are migrated to the server's Root tenant, even if initially created on an SCN linked to a second-level tenant.
- For telemetry migration:
  - Telemetry is imported separately for each node of the distributed solution, according to the scenario described below. The target node for telemetry can be selected during KUMA collector configuration.
- The following actions should be avoided until the export process is finished:
  - Add, edit or delete users
  - Perform any actions with operating system images in the Sandbox or modify Sandbox rules
  - Add, edit or delete prevention rules
  - Add, edit or delete IOC, IOA rules, YARA rules, IOA exclusions, and IOC scan schedule
  - Add or remove Endpoint Agents, or edit Endpoint Agent groups and tags
  - Change the list of archive passwords

Preparing for data migration

Before importing data, perform the following actions:

1. On the OSMP, create a tenant structure that meets the following criteria:
   - A separate tenant is created for each SCN, including Embedded SCNs.
   - SCN-corresponding tenants must not have parent-child relationships, as such relationships could distort the scope of the transferred objects.
   - For the PCN, as a unifying entity, a separate tenant must be created, which serves as the parent for the tenants corresponding to the SCN.
2. In the configuration file generated during data export (or at another time, but before the import), specify the mapping of Kaspersky Next XDR Expert nodes and tenants to OSMP tenants. The mapping must be **one-to-one**: assigning multiple SCNs to the same OSMP tenant is not allowed.

Stages

Data migration in a distributed solution follows the stage sequence described in the scenario for migration to OSMP deployed on a single node, with the following additions and modifications.

1. The data export procedure must be performed on each node of the distributed solution. The order in which nodes are processed is arbitrary and can be chosen for operational convenience. However, all exported data from all nodes must be placed on the same external disk.
2. Export and import operations are carried out in three steps. Two alternative execution schemes are supported:
   - *Full export—full import*: export for the first step, export for the second step, export for the third step (telemetry), and then import for the first step, import for the second step, and import for the third step. This approach increases total system downtime but allows the Kaspersky Next XDR Expert servers to be released earlier.
   - *Interleaved mode*: export for the first step and import for the first step, export for the second step and import for the second step, and then export for the third and import for the third step. This approach minimizes system downtime, but the Kaspersky Next XDR Expert servers remain occupied longer than in the full export—full import scheme.

To perform data migration using the full export—full import scheme:

1. **Mounting the external disk to the first node of the distributed deployment**

   If you prefer to use a shared folder as an external storage, create it on a separate Linux server.step

   How to create a network share

   To create a Samba share:

   1. Install Samba:

      ```
      sudo apt update
      sudo apt install samba samba-common-bin
      ```

   2. Add a Samba user with creating home folder for this user. This folder will be the shared folder.

Please do not create a shared folder manually as the current user. You must do this using the command below:

```
sudo useradd -m -d /home/<SHARE_FOLDER> -s /bin/bash <USER_NAME>
```

3. Set a password and activate a user:

```
sudo smbpasswd -a <USER_NAME>
sudo smbpasswd -e <USER_NAME>
```

4. Add the following block to the configuration file /etc/samba/smb.conf:

```
[<SHARE_FOLDER>]
comment = Network share
path = /home/<SHARE_FOLDER>
read only = no
browseable = yes
guest ok = no
```

where:

- `SHARE_FOLDER` is the name of a shared folder.

5. Restart the service:

```
sudo systemctl restart smbd nmbd
```

To create a Windows Share (cmd/PowerShell):

1. Make sure the console is running as administrator.

2. Create a user and set a password:

```
net user <USER_NAME> <PASSWORD> /add /expires:never
```

3. Create a folder and share it:

```
mkdir <SHARE_FOLDER>
icacls <SHARE_FOLDER> /inheritance:r
icacls <SHARE_FOLDER> /grant "Everyone:(OI)(CI)F"
```

4. Create a shared resource for a user:

```
net share <SHARE_NAME>=<SHARE_FOLDER> /grant:<USER_NAME>,full
```

In Technical Support Mode, mount an external disk (physical or network share) to the server where KEDR 7.1 is deployed.

How to mount and unmount a physical disk

To mount a physical disk:

1. Connect a physical hard drive to the host and find out its system name.

2. Search for the disk by its size in the command output:

```
lsblk
sudo fdisk -l
```

Example of a found device: /dev/sdb1

3. Format the device to the Ext4 file system to prepare the disk for writing data:

   `sudo mkfs.ext4 /dev/<disk>`

4. To create a mount point, create an empty directory where the disk will be mounted:

   `sudo mkdir -p /mnt/external_disk`

5. Run the following command depending on the disk file system:

   `sudo mount /dev/<disk> /mnt/external_disk`

To unmount a physical disk after performing required operations, run the following command:

`sudo umount /mnt/external_disk`

We strongly recommend to run this command before physically disconnect the cable to avoid data loss.

How to mount and unmount a network share

This method is intended for SMB/CIFS (Windows Share/Samba). It is suitable for a one-time connection. The connection will be lost after a system reboot.

To mount a network share:

1. Download and install the following three packages:

   `libtalloc2_2.4.2-1~bpo12+1_amd64.deb`

   `libwbclient0_4.9.5+dfsg-5+deb10u3astra.ce4_amd64.deb`

   `cifs-utils_6.8-2+deb10u1+ci202212062128+astra2_amd64.deb`

   Do not modify the source.list file. To install software, download the required .deb packages from public repositories or from private repositories supported by your organization.

2. Create a mount point:

   `sudo mkdir -p <PATH_FOLDER>`

3. Mount a network share:

   `sudo mount -t cifs //<SERVER>/<SHARE_FOLDER> /<PATH_FOLDER> -o username=<NAME>,password=<PASSWORD>,vers=2.1`

   where:

   - `SERVER` is the server's IP address or its network name.
   - `SHARE_FOLDER` is the name of a shared folder.
   - `PATH_FOLDER` is the local path to the mount point.
   - `NAME` is the user name on Windows/Samba.

- `PASSWORD` is the user password.
- `2.1` is the recommended protocol version.

  Command example:

  ```
  sudo mount -t cifs //<IP address>/Share /mnt/share -o username=<user name>,password=<password>,vers=2.1
  ```

To unmount a network share, run the following command:

```
sudo umount <PATH_FOLDER>
```

After completing the migration, it is recommended to remove the cifs-utils package and its dependencies.

To delete the cifs-utils package and its dependencies, run the following command:

```
sudo apt remove --purge cifs-utils && sudo apt autoremove -y
```

2. **Placing scripts and files required for data migration on the KATA/KEDR server**

In Technical Support Mode, place the export script and TAR archive containing Docker image img_kata_osmp_migrate.tar on the KATA Server.

The data export script performs a three-step export:

1. Exporting data required to start OSMP and replicate the KEDR 7.1 operational state:
   - List of active (enabled) users
   - List of image names included in the Sandbox reference set
   - Sandbox custom rules (scanning on custom images)
   - Setting that enables or disables file scanning in the Sandbox
   - Prevention rules
   - IOC rules
   - Custom IOA rules
   - Custom YARA rules
   - Passwords for archives used during anti-virus and Sandbox scan
   - IOC scan schedule
   - List of Endpoint Agents
   - Endpoint Agent groups and tags (assets)
2. Historical data and additional settings:
   - Alert database
   - Additional data related to alerts

- Repository (forensics, manually downloaded files, and files previously obtained via the `get` task)
- Email notification templates

3. Telemetry data

3. **Running the export script for the first step**

   Run the export script for the first step (specify the `--step first` parameter when running the kata-osmp-export.sh script), specifying the address of the mounted disk as the target location for the exported data.

   The script exports IOA exclusions (excludes.json) to a separate file in a relevant format for Endpoint Agent policy, and allows to import them later. This file can be found in the ioa folder of exported data.

   The script will notify you upon successful completion of the first data export step.

4. **Preparing for the second step of data export**

   After successful completion of the first step of data export, perform the following actions:

   - Disconnect KEDR 7.1 from all external integrations.
   - Disconnect Endpoint Agents in policy settings by deleting Endpoint Agent certificates and enabling certificate authentication. After this, Endpoint Agents will not be able to connect to the server.

5. **Running the export script for the second step**

   Run the export script for the second step (specify the `--step second` parameter when running the kata-osmp-export.sh script), specifying the address of the mounted disk as the target location for the exported data.

   The script will notify you upon successful completion of the first data export step.

6. **Running the export script for the third step**

   If you want to transfer telemetry data, run the export script for the third step (specify the `--step third` parameter when running the kata-osmp-export.sh script).

   The telemetry data will be under the path migration-dir/nodes/[GUID of the node]/archive_source/elasticsearch/ in the .dump files.

7. **Repeating steps 1-6**

   Repeat steps 1-6 of this procedure on the remaining nodes, either by physically moving the external disk or by reconnecting it to each node in turn.

   Use the same migration folder for data from all other nodes.

8. **Linux server deployment**

   Set up a separate Linux server.

   Configure the integration between Linux server and OSMP.

   Install Docker on the Linux server.

   Place the import script and TAR archive containing Docker image img_kata_osmp_migration_importer.tar on the separate Linux server.

   If KEDR 7.1 has not yet been replaced by Kaspersky Next XDR Expert 2.0 and is still running, one of its nodes (for example, PCN) can be used as the required Linux server. In this case, there is no need to install Docker, as it is already installed on the KEDR 7.1.

9. **Mounting an external disk to the Linux server**

   If you are using separate Linux server, unmount the external disk from last node of KEDR 7.1 in Technical Support Mode and mount it to the Linux server.

10. **Creating a configuration file to define the mapping between KEDR nodes and OSMP tenants**

    Configuration file example

    Configuration file example:

    ```
    {
      "globalTenantName": "Root tenant",
      "tenants": [
        {
          "scnName": "embedded_scn",
          "xdrTenantName": "embedded_scn"
        },
        {
          "scnName": "scn",
          "xdrTenantName": "scn"
        },
        {
          "scnName": "scn_2",
          "xdrTenantName": "scn_2"
        }
      ]
    }
    ```

    The tenant named embedded_scn is required, it indicates a Primary Central Node (PCN).

11. **Running the import script**

    On the Linux server, run the import script.

12. **Manually importing data**

    Perform the following actions:

    - If your Endpoint Agents versions are lower than those compatible with Kaspersky Next XDR Expert 2.0, you need to update them.
    - **Import custom IOA exclusions via Endpoint Agents policies**.

      1. Open the archive containing data to migrate.
      2. Extract the ioa.json file containing exclusions.
      3. Import the extracted file to KES policies (KES for Windows, KES for Linux).

    - Connect Endpoint Agents to OSMP server via Kaspersky Security Center server.

13. **Migrating telemetry data**

    You can migrate telemetry data by using external storage or without external storage.

To perform data migration using the interleaved mode:

1. **Mounting the external disk to the first node of the distributed deployment**

   If you prefer to use a shared folder as an external storage, create it on a separate Linux server.

   How to create a network share

To create a Samba share:

1. Install Samba:

   `sudo apt update`

   `sudo apt install samba samba-common-bin`

2. Add a Samba user with creating home folder for this user. This folder will be the shared folder.

   Please do not create a shared folder manually as the current user. You must do this using the command below:

   `sudo useradd -m -d /home/<SHARE_FOLDER> -s /bin/bash <USER_NAME>`

3. Set a password and activate a user:

   `sudo smbpasswd -a <USER_NAME>`

   `sudo smbpasswd -e <USER_NAME>`

4. Add the following block to the configuration file /etc/samba/smb.conf:

   `[<SHARE_FOLDER>]`

   `comment = Network share`

   `path = /home/<SHARE_FOLDER>`

   `read only = no`

   `browseable = yes`

   `guest ok = no`

   where:

   - `SHARE_FOLDER` is the name of a shared folder.

5. Restart the service:

   `sudo systemctl restart smbd nmbd`

To create a Windows Share (cmd/PowerShell):

1. Make sure the console is running as administrator.

2. Create a user and set a password:

   `net user <USER_NAME> <PASSWORD> /add /expires:never`

3. Create a folder and share it:

   `mkdir <SHARE_FOLDER>`

   `icacls <SHARE_FOLDER> /inheritance:r`

   `icacls <SHARE_FOLDER> /grant "Everyone:(OI)(CI)F"`

4. Create a shared resource for a user:

   `net share <SHARE_NAME>=<SHARE_FOLDER> /grant:<USER_NAME>,full`

In Technical Support Mode, mount an external disk (physical or network share) to the server where KEDR 7.1 is deployed.

How to mount and unmount a physical disk

To mount a physical disk:

1. Connect a physical hard drive to the host and find out its system name.
2. Search for the disk by its size in the command output:

   `lsblk`

   `sudo fdisk -l`

   Example of a found device: /dev/sdb1
3. Format the device to the Ext4 file system to prepare the disk for writing data:

   `sudo mkfs.ext4 /dev/<disk>`
4. To create a mount point, create an empty directory where the disk will be mounted:

   `sudo mkdir -p /mnt/external_disk`
5. Run the following command depending on the disk file system:

   `sudo mount /dev/<disk> /mnt/external_disk`

To unmount a physical disk after performing required operations, run the following command:

`sudo umount /mnt/external_disk`

> We strongly recommend to run this command before physically disconnect the cable to avoid data loss.

How to mount and unmount a network share

This method is intended for SMB/CIFS (Windows Share/Samba). It is suitable for a one-time connection. The connection will be lost after a system reboot.

To mount a network share:

1. Download and install the following three packages:

   `libtalloc2_2.4.2-1~bpo12+1_amd64.deb`

   `libwbclient0_4.9.5+dfsg-5+deb10u3astra.ce4_amd64.deb`

   `cifs-utils_6.8-2+deb10u1+ci202212062128+astra2_amd64.deb`

   > Do not modify the source.list file. To install software, download the required .deb packages from public repositories or from private repositories supported by your organization.
2. Create a mount point:

   `sudo mkdir -p <PATH_FOLDER>`
3. Mount a network share:

   `sudo mount -t cifs //<SERVER>/<SHARE_FOLDER> /<PATH_FOLDER> -o username=<NAME>,password=<PASSWORD>,vers=2.1`

where:

- `SERVER` is the server's IP address or its network name.
- `SHARE_FOLDER` is the name of a shared folder.
- `PATH_FOLDER` is the local path to the mount point.
- `NAME` is the user name on Windows/Samba.
- `PASSWORD` is the user password.
- `2.1` is the recommended protocol version.

  Command example:

  ```
  sudo mount -t cifs //<IP address>/Share /mnt/share -o username=<user name>,password=<password>,vers=2.1
  ```

To unmount a network share, run the following command:

```
sudo umount <PATH_FOLDER>
```

After completing the migration, it is recommended to remove the cifs-utils package and its dependencies.

To delete the cifs-utils package and its dependencies, run the following command:

```
sudo apt remove --purge cifs-utils && sudo apt autoremove -y
```

2. **Placing scripts and files required for data migration on the KATA/KEDR server**

In Technical Support Mode, place the export script and TAR archive containing Docker image img_kata_osmp_migrate.tar on the KATA Server.

The data export script performs a three-step export:

1. Exporting data required to start OSMP and replicate the KEDR 7.1 operational state:
   - List of active (enabled) users
   - List of image names included in the Sandbox reference set
   - Sandbox custom rules (scanning on custom images)
   - Setting that enables or disables file scanning in the Sandbox
   - Prevention rules
   - IOC rules
   - Custom IOA rules
   - Custom YARA rules
   - Passwords for archives used during anti-virus and Sandbox scan

- IOC scan schedule
- List of Endpoint Agents
- Endpoint Agent groups and tags (assets)

2. Historical data and additional settings:
   - Alert database
   - Additional data related to alerts
   - Repository (forensics, manually downloaded files, and files previously obtained via the `get` task)
   - Email notification templates
3. Telemetry data

3. **Running the export script for the first step**

   Run the export script for the first step (specify the `--step first` parameter when running the kata-osmp-export.sh script), specifying the address of the mounted disk where the data will be exported as the target location for the exported data.

   The script exports IOA exclusions (excludes.json) to a separate file in a relevant format for Endpoint Agent policy, and allows to import them later. This file can be found in the ioa folder of exported data.

4. **Repeating steps 1-3**

   Repeat steps 1-3 of this procedure on the remaining nodes, either by physically moving the external disk or by reconnecting it to each node in turn.

   Use the same migration folder for data from all other nodes.

5. **Linux server deployment**

   Set up a separate Linux server.

   Configure the integration between the Linux server and OSMP.

   Install Docker on the Linux server.

   Place the import script and TAR archive containing Docker image img_kata_osmp_migration_importer.tar on the separate Linux server.

   If KEDR 7.1 has not yet been replaced by Kaspersky Next XDR Expert 2.0 and is still running, one of its nodes (for example, PCN) can be used as the required Linux server. In this case there is no need to install Docker, as it is already installed on the KEDR 7.1.

6. **Mounting an external disk to the Linux server**

   If you are using separate Linux server, unmount the external disk from KEDR 7.1 and in Technical Support Mode, mount it to the Linux server.

7. **Creating a configuration file to define the mapping between KEDR nodes and OSMP tenants**

   Configuration file example

Configuration file example:

```
{
  "globalTenantName": "Root tenant",
  "tenants": [
    {
      "scnName": "embedded_scn",
      "xdrTenantName": "embedded_scn"
    },
    {
      "scnName": "scn",
      "xdrTenantName": "scn"
    },
    {
      "scnName": "scn_2",
      "xdrTenantName": "scn_2"
    }
  ]
}
```

The tenant named embedded_scn is required, it indicates a Primary Central Node (PCN).

8. **Running the import script**

   On the Linux server, run the import script.

9. **Preparing for the second step of data export**

   After successful completion of the first step of data import, perform the following actions:

   - Disconnect KEDR 7.1 from all external integrations.
   - Disconnect Endpoint Agents in policy settings by deleting Endpoint Agent certificates and enabling certificate authentication. After this, Endpoint Agents will not be able to connect to the server.
   - If your Endpoint Agents versions are lower than those compatible with Kaspersky Next XDR Expert 2.0, you need to update them.
   - **Import custom IOA exclusions via Endpoint Agents policies**.

     1. Open the archive containing data to migrate.
     2. Extract the ioa.json file containing exclusions.
     3. Import the extracted file to KES policies (KES for Windows, KES for Linux).

   - Connect Endpoint Agents to OSMP server via Kaspersky Security Center server.

10. **Connecting the same external disk to the first node of the distributed solution**

    In Technical Support Mode, mount an external disk (physical or network share) again to the server where KEDR 7.1 is deployed.

11. **Running the export script for the second step**

    Run the export script for the second step (specify the `--step second` parameter).

12. **Running the export script for the third step**

    If you want to transfer telemetry data, run the export script for the third step (specify the `--step third` parameter when running the kata-osmp-export.sh script).

    The telemetry data will be under the path migration-dir/nodes/[GUID of the node]/archive_source/elasticsearch/ in the .dump files.

13. **Repeating steps 10-12**

    Repeat steps 10-12 of this procedure on the remaining nodes, either by physically moving the external disk or by reconnecting it to each node in turn.

14. **Running the import script for the second step on OSMP**

    If you are using separate Linux server, unmount the external disk from KEDR 7.1 and in Technical Support Mode, mount it to the Linux server.

    On the Linux server, run the import script for the second step.

    The script will notify you upon successful completion of the second step of data import.

15. **Migrating telemetry data**

    You can migrate telemetry data by using external storage or without external storage.

Global and local data

This section defines how global and local data entities are treated during the export and import process.

For local data:

- All local data from each SCN (including embedded SCNs) is exported to the external disk and then transferred into the OSMP tenant explicitly mapped to that SCN in the configuration file.
- The scope of local entities after migration:
  - For local entities (rules, tasks, etc.) that previously applied to the entire SCN—the scope becomes the OSMP tenant corresponding to this SCN.
  - For local entities that previously applied to specific hosts—the scope remains the same hosts.
- Global data propagated from the PCN to the SCN is not transferred due to its incompleteness.

For global data:

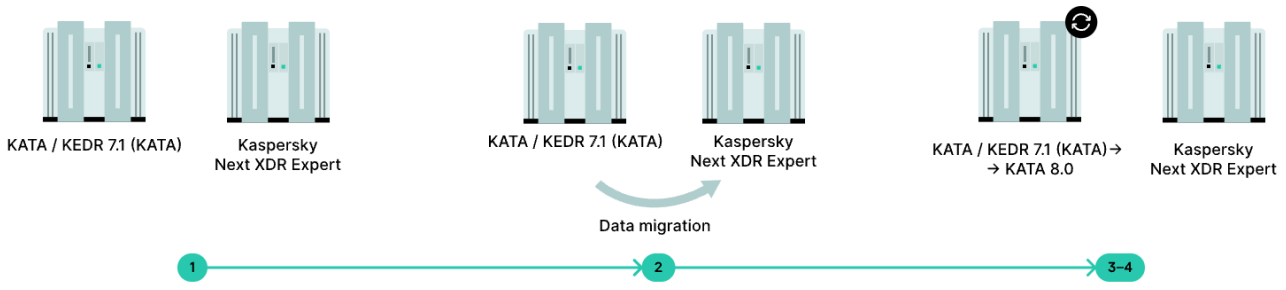
- All global data from each PCN is exported to the external disk and then transferred into the OSMP tenant explicitly mapped to that PCN in the configuration file.
- The scope of global entities after migration:
  - For global entities (rules, tasks, etc.) that previously applied to the entire tenant—the scope becomes all tenants corresponding to the SCNs that were part of the original tenant.
  - For global entities that previously applied to the specific SCN—the scope becomes the tenant corresponding to this SCN.
  - For global entities that previously applied to the specific hosts—the scope remains the same hosts.
- Local data received by the PCN from the SCNs is not transferred due to its incompleteness.

Scenario: Migration from KATA/KEDR 7.1 to separate KATA 8.0 and Kaspersky Next XDR Expert 2.0 solutions

This section covers the scenario for migration from a single KATA/KEDR 7.1 solution to two separate solutions: KATA 8.0 and platform Kaspersky Next XDR Expert 2.0.

Starting with Kaspersky Next XDR Expert 2.0, KEDR functionality becomes a part of Kaspersky Next XDR Expert, while KATA remains a separate solution. Therefore, to update the solution to the latest version, you must deploy a separate physical or virtual server to run Kaspersky Next XDR Expert.

The stages of migration are shown in the picture below.



Migration stages

Stages

Migration to separate solutions proceeds in stages:

1. **Kaspersky Next XDR Expert 2.0 deployment**

   Prepare the necessary equipment according to the sizing guide.

   Deploy Kaspersky Next XDR Expert 2.0 according to the instruction.

2. **Data migration from KATA 7.1 to Kaspersky Next XDR Expert 2.0**

   Migrate necessary data from KATA 7.1 to Kaspersky Next XDR Expert 2.0 using the appropriate scenario described in this section.

3. **KATA 7.1 upgrade**

   Upgrade KATA 7.1 to KATA 8.0 according to the instructions described in KATA documentation.

   > It is strongly recommended to backup data before upgrading. After the KATA upgrade, KEDR functionality and KEDR data will be unavailable.

4. **KATA resizing (optional)**

   If necessary, the sizing of KATA 8.0 can be increased as the solution requirements have been reduced by removing the KEDR functionality. For more information, refer to KATA sizing guide.

Switching Sandbox servers

Calculate the number of Sandbox servers required for KATA 8.0 functioning. Due to the migration of KEDR functionality to Kaspersky Next XDR Expert, the total number of servers needed should decreased.

Disconnect unnecessary Sandbox servers from KATA 7.1 and connect them to Kaspersky Next XDR Expert.

If necessary, deploy additional Sandbox servers for Kaspersky Next XDR Expert according the sizing guide.

Limitations for data migration in a cluster

- You can opt out of the clustered KATA deployment. To do this:
  1. Back up KATA 8.0 data.
  2. Deploy KATA 8.0 in a non-clustered configuration.
  3. Restore data from a backup.
- If the cluster includes three or more storage nodes, you can reduce the number of nodes to two. Please contact Technical Support for the instructions describing the steps to decommission the storage nodes.

Limitations for data migration in a distributed and multitenant solution

In a distributed and multitenant solution, follow the same steps as for a single-node deployment:

- Upgrade KATA to version 8.0 only after deploying all Kaspersky Next XDR Expert nodes and migrating all KEDR data.
- The migration script must first export data from all servers and only then import the data to Kaspersky Next XDR Expert.
- Phased (node-by-node) data migration is not supported.

Connecting Endpoint Agents to Kaspersky Next XDR Expert and telemetry collection server

This section describes how you can connect Endpoint Agents to Kaspersky Next XDR Expert and telemetry collection server.

There are two ways to connect Endpoint Agents:

- Connecting to the root tenant
- Connecting to child tenants

It is assumed that Endpoint Agents are connected via the current Kaspersky Security Center server.

Connecting Endpoint Agent to the root tenant

Connecting Endpoint Agent to Kaspersky Next XDR Expert

To connect Endpoint Agents to Kaspersky Next XDR Expert 2.0 root tenant, you need the tenant's address and the certificates for both the server and the Endpoint Agent.

In Kaspersky Next XDR Expert 2.0, follow these steps:

1. In the main menu, go to **Settings** → **Tenants**, select the **Root tenant**.
2. Go to **Settings** → **Certificates**.
3. Copy the server address from the **Agent Server FQDN** field.
   The server name should follow this formt: `agentserver.smp_domain`.
4. In the **Server certificate** section, click **Export** to export server certificate.
5. In the **Endpoint Agent certificate** section, click **Generate**.
6. Select the generated certificate and export it.
7. Specify and save the certificate password.

In Kaspersky Security Center, following these steps:

1. Open the Endpoint Agent policy and go to **Endpoint Detection and Response Expert (on-premise)** settings.
2. Select **Endpoint Detection and Response Expert (version 8.0 or later)**.
3. Go to the **Connection to response servers** section and click **Add** to enter the server address.
4. Go to **Connection settings** and import the previously exported server certificate.
5. Select **Use two way authentification**, import the previously exported Endpoint Agent certificate, and specify the certificate password.

Connecting Endpoint Agent to telemetry collection serve

To connect Endpoint Agent telemetry to the Kaspersky Next XDR Expert 2.0 root tenant, use the built-in [OOTB] Kaspersky EDR collector, and obtain the server address and certificates.

In Kaspersky Next XDR Expert 2.0, following these steps:

1. In the main menu, go to **Monitoring** → **Resources & services** → **Work with services** → **Active services**.
2. In the list of services, find the **[OOTB] Kaspersky EDR** service, select it and click **Deploy** in the menu.

3. In the opened window, copy the server address from the **API gateway URL** field.
   The server name should follow this format: `in.blh4zjhm4edckdt8qyprkpm2b.kuma_host.smp_domain`.
4. Close the window.
5. In the menu, click **Download certificate**.

In Kaspersky Security Center, follow these steps:

1. Open the Endpoint Agent policy and go to **Endpoint Detection and Response Expert (on-premise)** settings.
2. Select **Endpoint Detection and Response Expert (version 8.0 or later)**.
3. Go to the **Connection to telemetry collection servers** section and click **Add** to enter the server address.
4. Go to **Connection settings** and import the previously exported server certificate.

Connecting Endpoint Agent to child tenants

Connecting Endpoint Agent to Kaspersky Next XDR Expert

To connect Endpoint Agents to Kaspersky Next XDR Expert 2.0 child tenants, obtain the tenant addresses and certificates for both the servers and Endpoint Agents.

In Kaspersky Next XDR Expert 2.0, follow these steps:

1. In the main menu, go to **Settings** → **Tenants**, select the **Tenant**.
2. Go to **Settings** → **Certificates**.
3. Copy the server address from the **Agent Server FQDN** field.
   The server name should follow this format: `agentserver.smp_domain`.
4. In the **Server certificate** section, click **Export** to export server certificate.
5. In the **Endpoint Agent certificate** section, click **Generate**.
6. Select the generated certificate and export it.
7. Specify and save the certificate password.

In Kaspersky Security Center, follow these steps:

1. Open the Endpoint Agent policy and go to **Endpoint Detection and Response Expert (on-premise)** settings.
2. Select **Endpoint Detection and Response Expert (version 8.0 or later)**.
3. Go to the **Connection to response servers** section and click **Add** to enter the server address.

4. Go to **Connection settings** and import the previously exported server certificate.

5. Select **Use two way authentification**, import previously exported Endpoint Agent certificate and specify the certificate password.

Connecting Endpoint Agent to telemetry collection server

If you want to store telemetry events in separate tenants, you need to create and configure separate collectors and storages in each tenant, similar to the built-in ones.

In Kaspersky Next XDR Expert 2.0, follow these steps:

1. Create a collector based on the existing [OOTB] Kaspersky EDR:

   a. In the main menu, go to **Monitoring** → **Resources & services** → **Services configuration** → **Collectors**.

   b. Click the ••• button and select **Export resources**.

   c. In the **Tenant** drop-down list, select **Root tenant**.

   d. In the **File password** field, enter the password to protect the exported data.

   e. In the resource tree, select the check box for the following items:

   - **All resources** → **Collectors** → **KEDR** → **[OOTB] Kaspersky EDR**
   - **All resources** → **Correlators** → **[OOTB] OSMP Correlator**
   - **All resources** → **Storages** → **[OOTB] OSMP Storage**

   f. Click **Export**.

   g. Click the ••• button and select **Import resources**.

   h. In the **Tenant** drop-down list, select the required tenant.

   i. In the **Import source** drop-down list, select **File**.

   j. In the **File password** field, enter the password that you specified for the exported file.

   k. Click the **Select file** button and select the exported file.

   l. Click **Import**.

   Copy of the selected resources will appear for the tenant.

2. Deploy the created resources as services in the Kubernetes cluster:

   a. In the main menu, go to **Monitoring** → **Resources & services** → **Work with services** → **Active services**.

   b. Click **Add** and select **Add service**.

   c. In the opened window, select created resources and click **Create service**.

   New services corresponding to the selected resources will appear in the list of services.

d. Select the **Storage** service and click **Deploy**.

e. In the opened window, specify the required parameters and click **Deploy**.

f. Repeat steps a-e for each created service **strictly in the following order**: correlator, collector.

After creating a collector, ensure it is working correctly.

3. Obtain collector address and certificate:

   a. In the main menu, go to **Monitoring** → **Resources & services** → **Work with services** → **Active services**.

   b. In the list of services, find the created collector, select it and click **Deploy** in the menu.

   c. In the opened window, copy the server address in the **API gateway URL** field.
   The server name should follow this format: `in.blh4zjhm4edckdt8qyprkpm2b.kuma_host.smp_domain`.

   d. Close the window.

   e. In the menu, click **Download certificate**.

In Kaspersky Security Center, follow these steps:

1. Open the Endpoint Agent policy and go to **Endpoint Detection and Response Expert (on-premise)** settings.
2. Select **Endpoint Detection and Response Expert (version 8.0 or later)**.
3. Go to the **Connection to telemetry collection servers** section and click **Add** to enter the server address.
4. Go to **Connection settings** and import the previously exported server certificate.

Scenario: Telemetry migration using external storage

This section covers scenario for the telemetry data migration using an external storage.

If you want to migrate telemetry data, run the export script for the third step and specify the `--step third` parameter. Then set up a separate KUMA collector and import the previously exported telemetry as described below.

Do not deploy the KUMA collector to a Kubernetes cluster.

It is strongly recommended that you install the KUMA collector to a separate server and include this server in the KUMA cluster, which reduces the requirements for additional cores and CPUs.

The telemetry import process is time-consuming, but does not affect Kaspersky Next XDR Expert functionality.

Two separate telemetry migration schemes are supported:

- For a single-node deployment
- For a multitenancy mode

Telemetry migration for a single-node deployment

To migrate telemetry data for OSMP deployed on a single node:

1. Place Elasticsearch dump on the target server where the KUMA collector will be installed.

   The dump should be located in a path that meets the required criteria, for example: /mnt/elasticsearch.

2. Change the ownership of the dump:

   `sudo chown kuma:kuma <path to the dump>`

3. Run built-in KUMA collector **[OOTB] Kaspersky EDR events migration file**.

   Do not deploy the KUMA collector to a Kubernetes cluster.

   a. Go to **Monitoring** → **Resources & services** → **Services configuration** and click **Collectors**.

   b. In the list of collectors, find collector named **[OOTB] Kaspersky EDR events migration file**.

      The **Edit collector** wizard opens.

   c. Ensure that the **Kind** field in the **Transport** tab is set to **file**.

   d. In the **File path** field, specify the path with a mask to the dump files, for example: /mnt/elasticsearch/dump*.

   e. In the **Routing** tab, delete built-in **[OOTB] OSMP Storage**.

   f. Click **Add** and select **Create new** in the **Destination** field.

   g. Select **storage** in the **Kind** field.

   h. Select **[OOTB] OSMP Storage (Kubernetes gateway)** in the **URL** field.

      Enter the URL in the following format: https://api.d7fts6gikztq1geahwb7w12ha.kuma_host.smp_domain:443

      > Make sure that the record *.kuma_host.smp_domain exists and is properly configured on your DNS server.

   i. Click the **Save** button.

   j. At the setup validation step, click the **Create and save service** button.

      The recommended command will be displayed.

   k. Copy the recommended command and click **Save**.

   l. Run the recommended command with sudo on the server where KUMA services are installed, for example:

      ```
      sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the KUMA Core server>:<port used by KUMA
      Core for internal communication (port 7210 is used by default)> --id <service ID of collector copied from
      the web interface> --api.port <port used for communication with the installed component> --install
      ```

After creating the collector, make sure that it is working correctly.

After a while, telemetry history will be imported. You can check it by searching for events from created service.

Telemetry migration for a multitenancy mode

If Endpoint Agents were connected to different tenants during the telemetry connection step, the telemetry import must be performed into the same tenants configured to receive telemetry.

To migrate telemetry data for OSMP deployed in a multitenancy node:

1. Place Elasticsearch dumps on the target server where the KUMA collector will be installed.

   Dumps should be located in a path that meets the required criteria. For each KEDR 7.1, separate subfolders should be created, for example: /mnt/elasticsearch/pcn, /mnt/elasticsearch/scn1, /mnt/elasticsearch/scn2.

2. Change the ownership of dumps:

   ```
   sudo chown kuma:kuma <path to the dump>
   ```

3. To import telemetry into tenants, create and configure a separate collector in each tenant, similar to the built-in **[OOTB] Kaspersky EDR events migration file**. Storage for storing telemetry should already have been created in each tenant while connecting Endpoint Agents to child tenants.

   a. Go to **Monitoring** → **Resources & services** → **Services configuration** and click **Collectors**.

   b. Click the ••• button and select **Export resources**.

   c. In the **Tenant** drop-down list, select **Root tenant**.

   d. In the **File password** field, enter the password that must be used to protect exported data.

   e. In the resource tree, select the check box near **All resources** → **Collectors** → **KEDR** → **[OOTB] Kaspersky EDR events migration file**.

   f. Click **Export**.

   g. Click the ••• button and select **Import resources**.

   h. In the **Tenant** drop-down list, select the required tenant.

   i. In the **Import source** drop-down list, select **File**.

   j. In the **File password** field, enter the password that you specified for the exported file.

   k. Click the **Select file** button and select the exported file.

   l. Click **Import**.

   Copy of the selected collector will appear for the tenant.

4. Run the created collector.

Do not deploy the KUMA collector to a Kubernetes cluster.

a. Go to **Monitoring** → **Resources & services** → **Services configuration** and click **Collectors**.

b. In the list of collectors, find the created collector.

The **Edit collector** wizard opens.

c. Ensure that the **Kind** field in the **Transport** tab is set to **file**.

d. In the **File path** field, specify the path with a mask to the dump files, for example: /mnt/elasticsearch/scn1/dump*

e. In the **Routing** tab, delete built-in **[OOTB] OSMP Storage**.

f. Click **Add** and select **Create new** in the **Destination** field.

g. Select **storage** in the **Kind** field.

h. Select storage created for the current tenant in the **URL** field. There should be **Kubernetes gateway** note.

Enter the URL in the following format: https://api.d7fts6gikztq1geahwb7w12ha.kuma_host.smp_domain:443

> Make sure that the record *.kuma_host.smp_domain exists and is properly configured on your DNS server.

i. Click the **Save** button.

j. At the setup validation step, click the **Create and save service** button.

The recommended command will be displayed.

k. Copy the recommended command and click **Save**.

l. Run the recommended command with sudo on the server where KUMA services are installed, for example:

```
sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the KUMA Core server>:<port used by KUMA
Core for internal communication (port 7210 is used by default)> --id <service ID of collector copied from
the web interface> --api.port <port used for communication with the installed component> --install
```

After creating the collector, make sure that it is working correctly.

After a while, telemetry history will be imported. You can check it by searching for events from created service.

Scenario: Telemetry migration without external storage

This section covers scenario for the telemetry data migration without external storage. Telemetry migration requires two separate servers: KEDR 7.1 running on the KATA platform and Kaspersky Next XDR Expert 2.0 running on OSMP. The telemetry migration scenario below is part of the main Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using a separate server for Kaspersky Next XDR Expert 2.0.

Do not deploy the KUMA collector to a Kubernetes cluster.

It is strongly recommended that you install the KUMA collector to a separate server and include this server in the KUMA cluster, which reduces the requirements for additional cores and CPUs.

Limitations

Telemetry migration without external storage has the following limitation:

- If telemetry migration without using external storage is required, Kaspersky Next XDR Expert 2.0 cannot be deployed on the same server where KEDR 7.1 was deployed.

Stages

Telemetry migration proceeds in stages:

1. **Kaspersky Next XDR Expert deployment and data transfer**

   Perform steps 1-9 described in the Scenario: KATA/KEDR 7.1 to Kaspersky Next XDR Expert 2.0 migration using a separate server for Kaspersky Next XDR Expert 2.0.

2. **Preparing the KATA server for telemetry transfer by the KUMA collector and opening an external Elasticsearch port**

   Connect to the KATA server and perform the following steps in Technical Support Mode:

   1. Make sure that the following files exist on the KATA server:

      - /etc/kaspersky/node.json
      - /home/swarm_user/.ssh/id_rsa

   2. Run the following command to create the root directory for a script and other objects:

      ```
      mkdir /home/admin/elastic_script_dir
      ```

   3. Get the Docker image kata_osmp_migrate: copy a TAR archive containing Docker image kata_osmp_migrate to the server and run the following command:

      ```
      docker load -i ./img_kata_osmp_migrate.tar
      ```

      Copy the `Loaded image` value in the command output (for example, `registry.kata.zvx.com:5000/kaspersky/kata/management/kata_osmp_migrate:d0a217b`) and then use it in step 4 instead of `migrate_image:tag`.

   4. Run the image in the script generation mode:

      ```
      sudo docker run --rm \
          -v /home/admin/elastic_script_dir:/elastic_script_dir \
          -v /etc/kaspersky:/etc/kaspersky \
          -v /home/swarm_user/.ssh:/home/swarm_user/.ssh \
          --network kata_prod_network \
          migrate_image:tag \
          generate-elastic-script \
          --elastic-script-dir /elastic_script_dir
      ```

After generating the script, the console will display the data required for the KUMA collector:

- Elasticsearch endpoint
- Elasticsearch user
- Elasticsearch password
- Elasticsearch SSL certificate

SSL certificate will be located at /home/admin/elastic_script_dir/conf.d/elastic.crt.

Save the credentials securely, as you will not be able to copy or view them later.

To change the credentials, repeat step 4 of this guide.

5. Run the following command to go to the script directory:

```
cd /home/admin/elastic_script_dir
```

6. Run the script in the `open` mode:

```
sudo ./kata-elastic-access.sh open
```

After running this command, the external Elasticsearch port will be opened. The KATA server is ready for telemetry migration.

Elasticsearch will be available at https://<host>:8888/kata/elastic/

Elasticsearch indexes will be available at: https://<host>:8888/kata/elastic/_cat/indices?v&s=index:desc

3 Connecting KUMA collector to Elasticsearch

You can either create your own collector or use the built-in **[OOTB] Kaspersky EDR events migration Elasticsearch** collector.

We recommend to use the built-in collector, and the steps below explain how to use it.

1. In the main menu, go to **Monitoring** → **Resources & services** → **Services configuration** and click **Collectors**.
2. In the list of collectors, find collector named **[OOTB] Kaspersky EDR events migration Elasticsearch** and click on it.

   The **Edit collector** wizard opens.
3. On the **Transport** tab, in the **Connection** section, specify the following parameters:

   - **URL** is the URL of the Elasticsearch server (https://<KATA_server_ip>:8888/kata/elastic/).
   - **Index name** is the name of the index in Elasticsearch, that can be found at https://<KATA_server_ip>:8888/kata/elastic/_cat/indices?v&s=index:desc.
   - **Query** is the Elasticsearch query. We recommend setting the `size` parameter to no more than 10,000 to avoid performance issues with KUMA and Elasticsearch (for example, `"query" : { "match_all" : {} }, "size" : 10000, "sort": {"_doc" : "asc"}`.

- **Elastic credentials** is the secret used to store the credentials for connecting to the Elasticsearch server. To create a new secret, select **Create new** and specify credentials obtained at step 2.4 (*Run the image in the script generation mode*) of this procedure in the **User** and **Password** fields.

- **Elastic fingerprint** is the secret that stores secrets of the `fingerprint` type for connecting to the Elasticsearch server and secrets of the `certificate` type for using a CA certificate. To create a new secret, select **Create new** and for **Kind=certificate**, upload the elastic.crt certificate obtained at step 2.4 (*Run the image in the script generation mode*) of this procedure.

> You can create and use multiple connections in Connector to import data. Each connection corresponds to a single index in Elasticsearch. We recommend using up to five connections simultaneously. If you need to import more than five indexes, wait for the current import to finish in the collector, replace the indexes with the next set (up to five), update the Connector configuration, and run it again.
> As there can be many indexes, we recommend using the script to export and import telemetry using a separate external storage (specify the `--step third` parameter when running the script).

4. On the **Event parsing** tab, select the **[OOTB] Normalizer for KEDR events migration** normalizer.

5. In the **Routing** tab, delete built-in **[OOTB] OSMP Storage**.

6. Click **Add** and select **Create new** in the **Destination** field.

7. Select **storage** in the **Kind** field.

8. Select **[OOTB] OSMP Storage (Kubernetes gateway)** in the **URL** field.

   Enter the URL in the following format: https://api.d7fts6gikztq1geahwb7w12ha.kuma_host.smp_domain:443

   > Make sure that the record *.kuma_host.smp_domain exists and is properly configured on your DNS server.

9. Click the **Save** button.

10. At the setup validation step, click the **Create and save service** button.

    The recommended command will be displayed.

11. Copy the recommended command and click **Save**.

12. Run the recommended command with sudo on the server where KUMA services are installed, for example:

```
sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the KUMA Core server>:<port used by KUMA
Core for internal communication (port 7210 is used by default)> --id <service ID of collector copied from
the web interface> --api.port <port used for communication with the installed component> --install
```

After creating a collector, you are advised to make sure that it is working correctly. From this moment on, the telemetry transfer began with the parameters specified in the **Connection** section of the collector. You can check transferred telemetry data by searching it from created collector.

4 Closing access to Elasticsearch

After finishing import telemetry, run the script in the `close` mode to close access to Elasticsearch.

The steps described below must be performed on the KATA server in Technical Support Mode:

1. Run the following command to go to the script directory:

```
cd /home/admin/elastic_script_dir
```

2. Run the script in the close mode

```
sudo ./kata-elastic-access.sh close
```

Exporting data from KEDR 7.1

This section describes how you can create an archive containing data you want to transfer to Kaspersky Next XDR Expert 2.0.

The export is performed using the kata-osmp-export.sh script.

Running the export

Before running the export, make sure you have a TAR archive containing Docker image for export (for example, img_kata_osmp_migrate.tar).

To create an archive to export to Kaspersky Next XDR Expert 2.0:

1. Run the following command on the KATA host to make the kata-osmp-export.sh script executable:

```
chmod +x kata-osmp-export.sh
```

2. Run the kata-osmp-export.sh script:

```
sudo ./kata-osmp-export.sh --migration-dir migration_dir --migration-image <name of TAR archive containing
Docker image kata_osmp_migrate>.tar --step first --password <password for the archive> --log-dir <log
directory>
```

Where:

- `migration-dir` is the path to the migration directory where the data will be exported. This directory will later be used as input for the import command.
- `migration-image` is the TAR archive containing Docker image kata_osmp_migrate.

- `step` is an exporting step. Possible values:
  - `first`—settings and data will be exported: users, application settings, IOA rules, IOC, YARA, assets, and IOA exclusions.
  - `second`—historical data will be exported: alerts and repositories.
  - `third`—telemetry will be exported. For this step, you can specify a different `migration-dir` value than the one used in previous steps. This step is optional and can be performed if you want to migrate telemetry data using external storage. Note that telemetry data can also be migrated separately using another scenario.
- `password` is the password that will be used to encrypt the archive.
- `telemetry-max-processes` is the number of processes used for telemetry export, applicable for `step=third`. If not specified, this value defaults to the number of shards in Elasticsearch.

  The speed of telemetry export to external storage depends on the `telemetry-max-processes` parameter. We recommend setting this value to a multiple of four for a single node and eight for distributed deployments.

  In total, the telemetry export consumes an average of 1.5*`telemetry-max-processes value` cores. If this parameter is set to 16 or higher, the expected export performance is 4TB of Elasticsearch data per day.

  Estimated telemetry export time (days)

Number of days	Number of Endpoint Agents				
1000	3000	5000	10000	15000	
7	10	16	18	20	20
14	16	26	30	34	34
30	28	50	58	68	68

- `telemetry-keep-days` specifies the number of days of telemetry data to export, applicable for `step=third`. If not specified, all available telemetry data will be exported.
- `no-archive` is the optional parameter used to skip creating an archive. We recommend using this parameter for the third step to reduce export time. If this parameter is specified, the `password` parameter can be omitted.
- `log-dir` is the directory where export logs will be stored. If the directory does not exist, it will be created automatically. The migration.log file, containing the general export log, will be saved in this directory.

When exporting in multitenancy mode (for several nodes), each archive must be encrypted with the same password to ensure the import process succeeds.

Example:
To perform a full data export from a KATA node, run the script for step=first, step=second, and step=third:

```
sudo ./kata-osmp-export.sh --migration-dir migration_dir --migration-image <name of TAR archive containing Docker image kata_osmp_migrate>.tar --
step first --password <password for the archive> --log-dir <log directory>
sudo ./kata-osmp-export.sh --migration-dir migration_dir --migration-image <name of TAR archive containing Docker image kata_osmp_migrate>.tar --
step second --password <password for the archive> --log-dir <log directory>
sudo ./kata-osmp-export.sh --migration-dir telemetry_dir --migration-image <name of TAR archive containing Docker image kata_osmp_migrate>.tar --
step third --no-archive [--telemetry-max-processes <number of processes> --telemetry-keep-days <number of days>]
```

Preparing to run the import

This section describes how to prepare data exported from KEDR 7.1 for transfer to Kaspersky Next XDR Expert 2.0.

Preparation for the data import is performed in the following stages:

1. **Kaspersky Next XDR Expert 2.0 deployment**

   Before starting the import, ensure Kaspersky Next XDR Expert 2.0 is deployed where the import will be performed.

2. **cnab gateway-migration installation**

   Run the following command to check the availability of cnab gateway-migration and its version on the Kaspersky Next XDR Expert server:

   ```
   ./bin/kdt state -l gateway-migration
   ```

   If cnab gateway-migration is not installed and you only have the TAR archive, run the following command to install it:

   ```
   ./bin/kdt apply -k gateway-migration_<version>.tar
   ./bin/kdt invoke gateway-migration -a start
   ```

   The data import gateway will be launched, and three certificates required for running the import script will be created. These certificates must be saved to a separate Linux server (or KATA 7.1 server if it has not yet been replaced by Kaspersky Next XDR Expert 2.0 and is still running). The certificate files will be named as follows: `YYYY-MM-DDThh_mm_ss-ca.crt tls.crt tls.key`.

3. **Obtaining Kaspersky Next XDR Expert API token**

   Kaspersky Next XDR Expert API token is required to import specific entities, such as users.

   In the Kaspersky Next XDR Expert main menu, go to **Settings** → **API tokens**, click the **Add token** button, and specify a token name.

   In the **Expiration date** field, specify the token expiration date. This date must be set to the minimum duration necessary to complete the data migration.

   In the **API methods** section, select check boxes by the following methods:

   - **GET:/api/v1/tenants**
   - **GET:/api/v1/notifications/incident/template/default**
   - **POST:/api/v1/notifications/incident/template**
   - **POST:/api/v3/kuma/resources/*/create**
   - **POST:KEDR Migration**

   Click **Generate**, then **Copy and close**. The token will be created and copied to the clipboard. Save it securely, as you will not be able to copy it later.

4. **Forming migration-url**

Form the `migration-url` for the Kaspersky Next XDR Expert host domain name (FQDN). The corresponding record for the domain must be added to the DNS server.

The `migration-url` must start with "`https://migrate.`".

For example, for fqdn=example.com:

```
--migration-url https://migrate.example.com
```

5 **Adding tenants**

Add tenants to which you want to transfer data. For each node, a separate tenant must be added.

In the Kaspersky Next XDR Expert main menu, go to **Settings** → **Tenants**, select the **Root tenant**, click the **Add** button, specify a tenant name, and click **Add**.

In the **single-node mode**, specify the `--tenant-name` parameter in the command—the name of the tenant into which you want to import data. If the tenant name contains a space, it must be written in quotation marks, for example:

```
--tenant-name "Root tenant"
```

In the **multitenancy mode**, use the `--tenant-config-path` parameter in the command to specify a file containing information about the tenants exported from KATA and their corresponding tenants in Kaspersky Next XDR Expert for the import.

Configuration file example

Configuration file example:

```
{
  "globalTenantName": "Root tenant",
  "tenants": [
    {
      "scnName": "embedded_scn",
      "xdrTenantName": "embedded_scn"
    },
    {
      "scnName": "scn",
      "xdrTenantName": "scn"
    },
    {
      "scnName": "scn_2",
      "xdrTenantName": "scn_2"
    }
  ]
}
```

The tenant named embedded_scn is required, it indicates a Primary Central Node (PCN).

Importing data to Kaspersky Next XDR Expert 2.0

This section explains how to transfer data exported from KEDR 7.1 to Kaspersky Next XDR Expert 2.0.

The import is performed using the kata-osmp-import.sh script.

Before running the import, make sure you have a TAR archive containing Docker image kata_osmp_migration_importer, for example: img_kata_osmp_migration_importer.tar.

To import data to Kaspersky Next XDR Expert 2.0:

1. Run the following command to make the kata-osmp-import.sh script executable:

```
chmod +x kata-osmp-import.sh
```

2. In the **multitenancy mode**, run the following command on the host machine that contains the migration directory and has access to Kaspersky Next XDR Expert via a gateway:

```
./kata-osmp-import.sh --migration-image img_kata_osmp_migration_importer.tar --migration-dir
local_migration_dir --tenant-config-path tenant_config.json --migration-url https://migrate.example.com --
migration-token <API token> --password <archive password> --migration-key-file tls.key --migration-cert-file
tls.crt --migration-ca-cert-file 2026-02-10T05_46_06-ca.crt --log-dir <log directory> --log-level DEBUG
```

Where:

- `migration-image` is the TAR archive containing Docker image kata_osmp_migration_importer.
- `migration-dir` is the migration directory created during the export.
- `tenant-config-path` is the path to the configuration file containing information about tenants exported from KATA and the corresponding tenants in Kaspersky Next XDR Expert.
- `migration-url` is a URL required for data import.
- `migration-token` is a Kaspersky Next XDR Expert API token.
- `password` is a password used to encrypt archives in the migration directory.
- `migration-key-file`, `migration-cert-file`, `migration-ca-cert-file` are paths to certificates obtained and saved when starting gateway on the host.
- `ioa-exclude-output-path` is the path for saving IOA exclusions. This parameter is optional.

  IOA exclusions will not be migrated automatically. Instead, the exclusions_ioa.json file will be generated at the specified path. You can upload this file in the Kaspersky Next XDR Expert web console when **creating a policy**.

  To import IOA exclusions:

  1. In the main menu, go to **Asset management** → **Policies**.
  2. Select the administration group for which the policy is to be created:
     - For the root group.

       In this case you can proceed to the next step.
     - For a subgroup:

       a. Click the **Show hierarchy** button at the top of the page.

       b. The button name changes to **Hide hierarchy**.

       c. In the window that opens, click the name of the required subgroup.

The path to the selected group is displayed at the top of the page. If necessary, you can click a link with the administration group name to go to the group. By default, the last link in the path is inactive. Also, you can close the hierarchy window by clicking the **Hide hierarchy** button.

3. Click **Add**.

   The **Select application** window opens.

4. Select the application for which you want to create a policy.

5. If Kaspersky Security Network is supported by the Kaspersky application, for which you are creating a policy, you will be prompted to accept the KSN Statement. If you want to use Kaspersky Security Network, accept the KSN Statement.

6. Click **Next**.

   The new policy settings window opens with the **General** tab selected. If you want, change the default name, default status, and default inheritance settings of the policy.

7. Select the **Application settings** tab, expand the **General settings** menu on the left and select **Telemetry Settings**.

8. On the **EDR Expert (on-premise) telemetry** tab, click the **Import** button.

9. In the **Import from file** pane, upload the exclusions_ioa.json file by dragging and dropping or by clicking **Upload a file**.

10. Select whether you want to add IOA exclusions from the exclusions_ioa.json file to existing entries or you want to overwrite existing entries, click **Save**.

11. Click **Save** to save the policy.

The policy will appear in the list of policies.

If you omit this step and do not specify the `ioa-exclude-output-path` parameter, you can later manually extract the exclusions_ioa.json file from the archive created during the export process.

You can also specify the Samba path in the following format:

`smb://login:password@server/share`

`smb://server/share`

- `log-dir` is the directory where the import logs will be stored. If the specified directory does not exist, it will be created automatically. The migration.log file containing the general log that is output during the import will appear in the directory.

- `log-level` is the logging level, for example, `DEBUG`, to obtain extended logs. The skipped_import.log file containing information about objects skipped due to errors will appear in the directory specified by the `log-dir` parameter.

3. In the **single-node mode**, run the following command on the host machine that contains the migration directory and has access to Kaspersky Next XDR Expert via a gateway:

```
./kata-osmp-import.sh --migration-image img_kata_osmp_migration_importer.tar --migration-dir
local_migration_dir --tenant-name "Root tenant" --migration-url https://migrate.example.com --migration-token
<API token> --password <archive password> --migration-key-file tls.key --migration-cert-file tls.crt --
migration-ca-cert-file 2026-02-10T05_46_06-ca.crt --log-dir <log directory> --log-level DEBUG
```

   Unlike the multitenancy mode, in this case you need to specify the `tenant-name` parameter instead of `tenant-config-path`.

   Descriptions for other parameters given in step 2 of this instruction are also relevant for the single-node mode.

4. Once the import process is finished, install the Network Agent and configure the Network Agent policy settings on the host.

When the migration procedure is started again, the data added after the previous migration is transferred.

> We strongly recommend that you do not delete any data in the migration directory (`migration-dir`).

General limitations

> We recommend migrating only alerts with **Closed** status

General limitations for migration are the following:

- Mail server integration settings for sending messages via SMTP are not transferred.
- Telemetry accumulated by Endpoint Agents during the period after they are disconnected from the Kaspersky Next XDR Expert server until they are connected to the new server will be irretrievably lost.
- The following actions are not recommended while the export process is in progress:
  - Add, edit or delete users.
  - Perform any actions with operating system images in the Sandbox or modifying Sandbox rules.
  - Add, edit or delete prevention rules.
  - Add, edit or delete IOC, IOA rules, YARA rules, IOA exclusions, and IOC scan schedule.
  - Add or remove Endpoint Agents, or edit Endpoint Agent groups and tags.
  - Change the list of archive passwords.
- While being transferred, some data is permanently lost, specifically:
  - Alert history.
  - Completed tasks — Note that if task results are saved to storage, they remain intact.

- Inactive users and user accounts associated with Active Directory.
- Other data not included in the transferred data list.

- Due to OSMP limitations, during the migration of data from a distributed or multitenant solution, data may be altered or lost, specifically:
  - Global rules and settings defined on the PCN may be pushed down to the tenant hierarchy.
  - In case of conflicts between settings or rules transferred from PCN and SCN, the SCN settings and rules may not be imported into OSMP.

    For example, if the same blocking rule exists on both PCN and SCN, whether the SCN rule is transferred depends on whether the PCN and SCN scopes overlap.

    Information about all data that failed to transfer is logged.

Setting up and working with the SIEM component

This chapter provides information about installing and configuring the KUMA SIEM system.

KUMA services

Services are the main components of KUMA that help the system to manage events: services allow you to receive events from event sources and subsequently bring them to a common form that is convenient for finding correlation, as well as for storage and manual analysis. Each service consists of two parts that work together:

- One part of the service is created inside the KUMA Console based on set of resources for services.
- The second part of the service is installed in the network infrastructure where the KUMA system is deployed as one of its components. The server part of a service can consist of multiple instances: for example, services of the same agent or storage can be installed on multiple devices at once.

  On the server side, KUMA services are located in the `/opt/kaspersky/kuma` directory.

  When you install KUMA in high availability mode, only the KUMA Core is installed in the cluster. Collectors, correlators, and storages are hosted on hosts outside of the Kubernetes cluster.

Parts of services are connected to each other via the service ID.

Service types:

- Storages are used to save events.
- Correlators are used to analyze events and search for defined patterns.
- Collectors are used to receive events and convert them to KUMA format.
- Agents are used to receive events on remote devices and forward them to KUMA collectors.

In the KUMA Console, services are displayed in the **Resources** → **Active services** section in table format. The table of services can be updated by clicking the **Refresh** button and sorted by columns by clicking on the active headers.

The maximum table size is not limited. If you want to select all services, scroll to the end of the table and select the **Select all** check box, which selects all available services in the table.

Table columns:

- **Status**—service status:
  - Green means that the service is running.
  - Red means that the service is not running.

- Yellow is the status that applies to all services except the agent. The yellow status means that the service is running, but there are errors or alerts from Victoria Metrics. You can view the error message by hovering the mouse pointer over the status.

- Purple is the status that is applied to running services whose configuration file in the database has changed, but that have no other errors. If a service has an incorrect configuration file and has errors, for example, from Victoria Metrics, status of the service is yellow.

- Gray—if a deleted tenant had a running service that continues to work, that service is displayed with a gray status on the **Active services** page. Services with the gray status are kept to let you copy the ID and remove services on your servers. Only the Main administrator can delete services with the gray status. When a tenant is deleted, the services of that tenant are assigned to the Main tenant.

- **Type**—type of service: **agent**, **collector**, **correlator**, or **storage**.
- **Name**—name of the service. Clicking on the name of the service opens its settings.
- **Version**—service version.
- **Tenant**—the name of the tenant that owns the service.
- **FQDN**—fully qualified domain name of the service server.
- **IP address**—IP address of the server where the service is installed.
- **API port**—Remote Procedure Call port number.
- **Uptime**—the time showing how long the service has been running.
- **Created**—the date and time when the service was created.

The table can be sorted in ascending and descending order, as well as by the **Status** parameter. To sort active services, right-click the context menu and select one or more statuses.

You can click the buttons in the upper part of the **Services** window to perform the following group actions:

- Add service

  You can create new services based on existing service resource sets. We do not recommend creating services outside the main tenant without first carefully planning the inter-tenant interactions of various services and users.

- Refresh

  You can refresh the list of active services.

- Update configuration
- Restart

To perform an action with an individual service, right-click the service to display its context menu. The following actions are available:

- Reset certificate
- Delete

- Download log

  If you want to receive detailed information, enable the Debug mode in the service settings.

- Copy service ID

  You need this ID to install, restart, stop, or delete the service.

- Go to Events

- Go to active lists

- Go to context tables

- Go to partitions

To change a service, select a service under **Resources** → **Active services**. This opens a window with a set of resources based on which the service was created. You can edit the settings of the set of resources and save your changes. To apply the saved changes, restart the service.

> If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Resources** → **Normalizers** in the web interface.

Services tools

This section describes the tools for working with services available in the **Resources** → **Active services** section of the KUMA Console.

Getting service identifier

The service identifier is used to bind parts of the service residing within KUMA and installed in the network infrastructure into a single complex. An identifier is assigned to a service when it is created in KUMA, and is then used when installing the service to the server.

To get the identifier of a service:

1. Log in to the KUMA Console and open **Resources** → **Active services**.

2. Select the check box next to the service whose ID you want to obtain, and click **Copy ID**.

   The identifier of the service will be copied to the clipboard. It can be used, for example, for installing the service on a server.

Stopping, starting, checking status of the service

While managing KUMA, you may need to perform the following operations.

- Temporarily stop the service. For example, when restoring the Core from backup, or to edit service settings related to the operating system.
- Start the service.
- Check the status of the service.

The table below lists commands that may be useful when managing KUMA.

Commands for stopping, starting, and checking the status of a service

Service	Stop service	Start service	Check the status of the service
Core	`sudo systemctl stop kuma-core.service`	`sudo systemctl start kuma-core.service`	`sudo systemctl status kuma-core.service`
Services with an ID: • collector • correlator • storage	`sudo systemctl stop kuma-<collector/correlator/storage>-<service ID>.service`	`sudo systemctl start kuma-<collector/correlator/storage>-<service ID>.service`	`sudo systemctl status kuma-<collector/correlator/storage><service ID>.service`
Services without an ID: • kuma-grafana.service • kuma-mongodb.service • kuma-victoria-metrics.service • kuma-vmalert.service	`sudo systemctl stop kuma-<grafana/victoria-metrics/vmalert>.service`	`sudo systemctl start kuma-<grafana/victoria-metrics/vmalert>.service`	`sudo systemctl status kuma-<grafana/victoria-metrics/vmalert>.service`
Windows agents	To stop an agent service: 1. Copy the agent ID in the KUMA Console. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as a user with administrative privileges. 4. Run the following command in PowerShell: `Stop-Service -Name "WindowsAgent-<agent ID>"`	To start an agent service: 1. Copy the agent ID in the KUMA Console. 2. Connect to the host on which you want to start the KUMA agent service. 3. Run PowerShell as a user with administrative privileges. 4. Run the following command in PowerShell: `Start-Service -Name "WindowsAgent-<agent ID>"`	To view the status of an agent service: 1. In Windows, go to the Start → Services menu, and in the list of services, double-click the relevant KUMA agent. 2. This opens a window; in that window, view the status of the agent in the Service status field.

Restarting the service

To restart the service:

1. Log in to the KUMA web interface and open **Resources** → **Active services**.

2. Select the check box next to the service and select the necessary option:

   - **Update configuration**—perform a hot update of a running service configuration. For example, you can change the field mapping settings or the destination point settings this way.

   - **Restart**—stop a service and start it again. This option is used to modify the port number or connector type.

     Restarting KUMA agents:

     - KUMA Windows Agent can be restarted as described above only if it is running on a remote computer. If the service on the remote computer is inactive, you will receive an error when trying to restart from KUMA. In that case you must restart KUMA Windows Agent service on the remote Windows machine. For information on restarting Windows services, refer to the documentation specific to the operating system version of your remote Windows computer.

     - KUMA Agent for Linux stops when this option is used. To start the agent again, you must execute the command that was used to start it.

   - **Reset certificate**—remove certificates that the service uses for internal communication. This option may not be used to renew the Core certificate. To renew KUMA Core certificates, they must be reissued.

     Special considerations for deleting Windows agent certificates:

     - If the agent has the green status and you select **Reset certificate**, KUMA deletes the current certificate and creates a new one, the agent continues working with the new certificate.

     - If the agent has the red status and you select **Reset certificate**, KUMA generates an error that the agent is not running. In the %PROGRAMDATA%\Kaspersky Lab\KUMA\agent\<agent ID>\certificates folder, manually delete the internal.cert and internal.key files and start the agent manually. When the agent starts, a new certificate is created automatically.

     Special considerations for deleting Linux agent certificates:

     1. Regardless of the agent status, apply the **Reset certificate** option in the web interface to delete the certificate in the databases.

     2. In the agent installation folder, /opt/kaspersky/agent/<Agent ID>/certificates, manually delete the internal.cert and internal.key files.

     3. Since the **Reset certificate** option stops the agent, to continue its operation, start the agent manually. When the agent starts, a new certificate is created automatically.

Deleting the service

Before deleting the service get its ID. The ID will be required to remove the service for the server.

To remove a service in the KUMA Console:

1. Log in to the KUMA Console and open **Resources** → **Active services**.
2. Select the check box next to the service you want to delete, and click **Delete**.

   A confirmation window opens.
3. Click **OK**.

The service has been deleted from KUMA.

To remove a service from the server, run the following command:

```
sudo /opt/kaspersky/kuma/kuma <collector/correlator/storage> --id <service ID> --
uninstall
```

```
The service has been deleted from the server.
```

Viewing the table of active queries

A General Administrator can view the queries of other users and KUMA resources that are currently being executed.

To view the table of active queries:

1. In the main application window, go to the **Monitoring** → **Resources and services** → **Active services** section.

2. Select the check boxes next to the relevant storages.

3. In the toolbar or in the menu (⋮), click **Active queries**.

   This opens the window with the **ClickHouse active queries** table. When selecting multiple storages, the table displays active queries from each selected storage.

The resource table contains the following columns:

- **Started** is the date and time when the query was initiated.

- **Created by** is the initiator of the query. For user queries, the user name is displayed. For queries of the **Resource** type, the resource name is displayed.

- **Query** is the text of the SQL query executed in ClickHouse.

- **Type** is the type of the query. The following query types are available:

  - **In background** indicates that the query was initiated by the user in the background using the **Run in background** button.

  - **Resource** indicates an internal SQL query initiated by a KUMA resource, for example, during data processing or analysis.

  - **User** indicates an SQL query initiated by the user using the **Run query** button and executed normally.

- **Storage** is the name of the ClickHouse storage in whose context the query is being executed.

Click the ⚙ button to configure the displayed columns of the queries table. If necessary, you can force the update of the data by clicking ⟳.

If you want to sort the queries by a specific column, click its title and select one of the available options in the drop-down list: **Ascending** or **Descending**.

If necessary, you can cancel queries.

To cancel a query:

1. Open the **ClickHouse active queries** table.
2. Select the check box next to a query and click **Cancel query**.

   A confirmation window opens. You can also specify the reason for canceling the query in the **Specify a reason** field.
3. Click **OK**.

The query is canceled and removed from the table. An audit event is automatically generated.

Partitions window

If the storage service was created and installed, you can view its partitions in the **Partitions** table.

*To open **Partitions** table:*

1. Log in to the KUMA Console and open **Resources** → **Active services**.
2. Select the check box next to the relevant storage and click **Go to partitions**.

The **Partitions** table opens.

The table has the following columns:

- **Tenant**—the name of the tenant that owns the stored data.
- **Created**—partition creation date.
- **Space**—the name of the space.
- **Size**—the size of the space.
- **Events**—the number of stored events.
- **Transfer to cold storage**—the date when data will be migrated from the ClickHouse clusters to cold storage disks.
- **Expires**—the date when the partition expires. After this date, the partition and the events it contains are no longer available.

You can delete partitions.

To delete a partition:

1. Open the **Partitions** table (see above).
2. Select the check box to the left from the required partition.

3. Select **Delete**.

   A confirmation window opens.

4. Click **OK**.

The partition has been deleted. Audit event partitions cannot be deleted.

Searching for related events

You can search for events processed by the Correlator or the Collector services.

To search for events related to the Correlator or the Collector service:

1. Log in to the KUMA Console and open **Resources** → **Active services**.

2. Select the check box next to the required correlator or collector and click **Go to Events**.

   A new browser tab opens with the KUMA **Events** section open.

3. To find events, click the icon.

   A table with events selected by the search expression `ServiceID = <ID of the selected service>` will be displayed.

Service resource sets

Service resource sets are a resource type, a KUMA component, a set of settings based on which the KUMA services are created and operate. Resource sets for services are collections of resources.

Any resources added to a set of resources must be owned by the same tenant that owns the created set of resources. An exception is the shared tenant, whose owned resources can be used in the sets of resources of other tenants.

Resource sets for services are displayed in the **Resources** → **<Resource set type for the service>** section of the KUMA Console. Available types:

- Collectors
- Correlators
- Storages
- Agents

When you select the required type, a table opens with the available sets of resources for services of this type. The resource table contains the following columns:

- **Name**—the name of a resource set. Can be used for searching and sorting.
- **Updated**—the date and time of the last update of the resource set. Can be used for sorting.

- **Created by**—the name of the user who created the resource set.
- **Description**—the description of the resource set.

Creating a storage

A storage consists of two parts: one part is created inside the KUMA Console, and the other part is installed on network infrastructure servers intended for storing events. The server part of a KUMA storage consists of ClickHouse nodes collected into a cluster. ClickHouse clusters can be supplemented with cold storage disks.

For each ClickHouse cluster, a separate storage must be installed.

> Prior to storage creation, carefully plan the cluster structure and deploy the necessary network infrastructure. When choosing a ClickHouse cluster configuration, consider the specific event storage requirements of your organization.

> It is recommended to use ext4 as the file system.

A storage is created in several steps:

1. **Creating a set of resources for a storage in the KUMA Console**
2. **Creating a storage service in the KUMA Console**
3. **Installing storage nodes in the network infrastructure**

> When creating storage cluster nodes, verify the network connectivity of the system and open the ports used by the components.

If the storage settings are changed, the service must be restarted.

ClickHouse cluster structure

A ClickHouse *cluster* is a logical group of devices that possess all accumulated normalized KUMA events. It consists of one or more logical *shards*.

A *shard* is a logical group of devices that possess a specific **portion** of all normalized events accumulated in the cluster. It consists of one or more *replicas*. Increasing the number of shards lets you do the following:

- Accumulate more events by increasing the total number of servers and disk space.
- Absorb a larger **stream** of events by distributing the load associated with an influx of new events.
- Reduce the time taken to search for events by distributing search zones among multiple devices.

A *replica* is a device that is a member of a logical shard and possesses a single copy of that shard's data. If multiple replicas exist, it means multiple copies exist (the data is replicated). Increasing the number of replicas lets you do the following:

- Improve high availability.
- Distribute the total load related to data searches among multiple machines (although it is best to increase the number of shards for this purpose).

A *keeper* is a device that participates in **coordination** of data replication at the **whole** cluster level. At least one device per cluster must have this role. The recommended number of the devices with this role is 3. The number of devices involved in coordinating replication must be an **odd** number. The *keeper* and *replica* roles can be combined in one machine.

ClickHouse cluster node settings

Prior to storage creation, carefully plan the cluster structure and deploy the necessary network infrastructure. When choosing a ClickHouse cluster configuration, consider the specific event storage requirements of your organization.

When creating ClickHouse cluster nodes, verify the network connectivity of the system and open the ports used by the components.

For each node of the ClickHouse cluster, you need to specify the following settings:

- Fully qualified domain name (FQDN)—a unique address to access the node. Specify the entire FQDN, for example, `kuma-storage.example.com`.
- Shard, replica, and keeper IDs—the combination of these settings determines the position of the node in the ClickHouse cluster structure and the node role.

Node roles

The roles of the nodes depend on the specified settings:

- shard, replica, keeper—the node participates in the accumulation and search of normalized KUMA events and helps coordinate data replication at the cluster-wide level.
- shard, replica—the node participates in the accumulation and search of normalized KUMA events.
- keeper—the node **does not** accumulate normalized events, but helps coordinate data replication at the cluster-wide level. Dedicated keepers must be specified at the beginning of the list in the **Resources** → **Storages** → <Storage> → **Basic settings** → **ClickHouse cluster nodes** section.

ID requirements:

- If multiple shards are created in the same cluster, the shard IDs must be unique within this cluster.
- If multiple replicas are created in the same shard, the replica IDs must be unique within this shard.
- The keeper IDs must be unique within the cluster.

Example of ClickHouse cluster node IDs:

- shard 1, replica 1, keeper 1;
- shard 1, replica 2;
- shard 2, replica 1;
- shard 2, replica 2, keeper 3;
- shard 2, replica 3;
- keeper 2.

Cold storage of events

In KUMA, you can configure the migration of legacy data from a ClickHouse cluster to the cold storage. Cold storage can be implemented using the local disks mounted in the operating system, a Hadoop Distributed File System (HDFS), or an S3 cloud storage (Simple Storage Service).

Starting with KUMA version 4.2, data stored on HDFS disks will be gradually migrated to S3 cloud storage. Support for the HDFS file system will be discontinued in a future release of the application.

Cold storage is enabled when at least one cold storage disk is specified. If multiple storages are used, a cold storage disk or a HDFS or S3 disk must be mounted at the path specified in the storage configuration on each node with data. If a cold storage disk is not configured and the server runs out of disk space in hot storage, the storage service is stopped. If both hot storage and cold storage are configured, and space runs out on the cold storage disk, the KUMA storage service is stopped. We recommend avoiding such situations by adding custom event storage conditions in hot storage.

Cold storage disks can be added or removed. If you have added multiple cold storage disks, data is written to them in a round-robin manner. If data to be written to disk would take up more space than is available on that disk, this data and all subsequent data is written round-robin to the next cold storage disks. If you added only two cold storage disks, the data is written to the drive that has free space left.

After changing the cold storage settings, the storage service must be restarted. If the service does not start, the reason is indicated in the storage log.
Restarting the service is also necessary if the path to the S3 storage is specified incorrectly. Note that the created *storage.xml* file is not automatically deleted. You need to manually clear the `/opt/kaspersky/kuma/clickhouse/cfg/config.d/` folder and configure cold storage again.

If the cold storage disk specified in the storage settings has become unavailable (for example, out of order), this may lead to errors in the operation of the storage service. In this case, recreate a disk with the same path (for local disks) or the same address (for HDFS and S3 disks) and then delete it from the storage settings.

Rules for moving the data to the cold storage disks

You can configure the storage conditions for events in hot storage of the ClickHouse cluster by setting a limit based on retention time or on maximum storage size. When cold storage is used, every 15 minutes and after each Core restart, KUMA checks if the specified storage conditions are satisfied:

1. KUMA gets the partitions for the storage being checked and groups the partitions by cold storage disks and spaces.

2. For each space, KUMA checks whether the specified storage condition is satisfied.

3. If the condition is satisfied (for example, if the space contains that exceed their retention time, or if the size of the storage has reached or exceeded limit specified in the condition), KUMA transfers all partitions with the oldest date to cold storage disks or deletes these partitions if no cold storage disk is configured or if it is configured incorrectly. This action is repeated while the configured storage condition remains satisfied in the space; for example, if after deleting partitions for a date, the storage size still exceeds the maximum size specified in the condition.

   KUMA generates audit events when data transfer starts and ends, or when data is removed.

4. If retention time is configured in KUMA, whenever partitions are transferred to cold storage disks, it is checked whether the configured conditions are satisfied on the disk. If events are found on the disk that have been stored for longer than the **Event retention time**, which is counted from the moment the events were received in KUMA, the solution deletes these events or all partitions for the oldest date.

   KUMA generates audit events when it deletes data.

If the ClickHouse cluster disks are 95% full, the biggest partitions are automatically moved to the cold storage disks. This can happen more often than once per hour.

During data transfer, the storage service remains operational, and its status stays green in the **Resources** → **Active services** section of the KUMA web interface. When you hover over the status icon, a message is displayed about the data transfer. When a cold storage disk is removed, the storage service has the yellow status.

Special considerations for storing and accessing events

- When using HDFS disks for cold storage, protect your data in one of the following ways:
  - Configure a separate physical interface in the VLAN, where only HDFS disks and the ClickHouse cluster are located.
  - Configure network segmentation and traffic filtering rules that exclude direct access to the HDFS disk or interception of traffic to the disk from ClickHouse.
- Events located in the ClickHouse cluster and on the cold storage disks are equally available in the KUMA web interface. For example, when you search for events or view events related to alerts.
- You can disable the storage of events or audit events on cold storage disks. To do so, specify the following in storage settings:
  - If you do not want to store events on cold storage disks, do one of the following:
    - If in the **Retention policy options** field, you have selected a gigabyte or percentage based storage condition, in the **Events storage time** field, specify `0`.
    - If in the **Retention policy options** field, you have selected a storage condition in days, in the **Event storage time** field, specify the same number of days as in the **Retention policy options**.
  - If you do not want to store audit events on cold storage disks, in the **Audit cold retention period** field, specify `0` (days).

Special considerations for using HDFS disks

- Before connecting HDFS disks, create directories for each node of the ClickHouse cluster on them in the following format: `<HDFS disk host>/<shard ID>/<replica ID>`. For example, if a cluster consists of two nodes containing two replicas of the same shard, the following directories must be created:

  - hdfs://hdfs-example-1:9000/clickhouse/1/1/

  - hdfs://hdfs-example-1:9000/clickhouse/1/2/

  Events from the ClickHouse cluster nodes are migrated to the directories with names containing the IDs of their shard and replica. If you change these node settings without creating a corresponding directory on the HDFS disk, events may be lost during migration.

- HDFS disks added to storage operate in the JBOD mode. This means that if one of the disks fails, access to the storage will be lost. When using HDFS, take high availability into account and configure RAID, as well as storage of data from different replicas on different devices.

- The speed of event recording to HDFS is usually lower than the speed of event recording to local disks. The speed of accessing events in HDFS, as a rule, is significantly lower than the speed of accessing events on local disks. When using local disks and HDFS disks at the same time, the data is written to them in turn.

- HDFS is used only as distributed file data storage of ClickHouse. Compression mechanisms of ClickHouse, not HDFS, are used to compress data.

- The ClickHouse server must have write access to the corresponding HDFS storage.

Removing cold storage disks

Before physically disconnecting cold storage disks, remove these disks from the storage settings.

To remove a disk from the storage settings:

- In the KUMA Console, under **Resources** → **Storages**, select the relevant storage.

  This opens the storage.

- In the window, in the **Disks for cold storage** section, in the required disk's group of settings, click **Delete disk**.

  Data from removed disk is automatically migrated to other cold storage disks or, if there are no such disks, to the ClickHouse cluster. During data migration, the storage status icon is highlighted in yellow. Audit events are generated when data transfer starts and ends.

- After event migration is complete, the disk is automatically removed from the storage settings. It can now be safely disconnected.

Removed disks can still contain events. If you want to delete them, you can manually delete the data partitions using the DROP PARTITION command.

If the cold storage disk specified in the storage settings has become unavailable (for example, out of order), this may lead to errors in the operation of the storage service. In this case, create a disk with the same path (for local disks) or the same address (for HDFS disks) and then delete it from the storage settings.

Detaching, archiving, and attaching partitions

If you want to optimize disk space and speed up queries in KUMA, you can detach data partitions in ClickHouse, archive partitions, or move partitions to a drive. If necessary, you can later reattach the partitions you need and perform data processing.

Detaching partitions

To detach partitions:

1. Determine the shard on all replicas of which you want to detach the partition.

2. Get the partition ID using the following command:

   ```
   sudo /opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "SELECT
   partition, name FROM system.parts;" |grep 20231130
   ```

   In this example, the command returns the partition ID for November 30, 2023.

3. One each replica of the shard, detach the partition using the following command and specifying the partition ID:

   ```
   sudo /opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER
   TABLE events_local_v2 DETACH PARTITION ID '<partition ID>'"
   ```

As a result, the partition is detached on all replicas of the shard. Now you can move the data directory to a drive or archive the partition.

Archiving partitions

To archive detached partitions:

1. Find the detached partition in disk subsystem of the server:

   ```
   sudo find /opt/kaspersky/kuma/clickhouse/data/ -name <ID of the detached partition>\*
   ```

2. Change to the 'detached' directory that contains the detached partition, and while in the 'detached' directory, perform the archival:

   ```
   sudo cd <path to the 'detached' directory containing the detached partition>
   ```

   ```
   sudo zip -9 -r detached.zip *
   ```

   For example:

   ```
   sudo cd /opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-
   d8d8e1eb3968/detached/
   ```

   ```
   sudo zip -9 -r detached.zip *
   ```

The partition is archived.

Attaching partitions

To attach archived partitions to KUMA:

1. Increase the **Retention period** value.

KUMA deletes data based on the date specified in the Timestamp field, which records the time when the event is received, and based on the **Retention period** value that you set for the storage.

Before restoring archived data, make sure that the **Retention period** value overlaps the date in the Timestamp field. If this is not the case, the archived data will be deleted within 1 hour.

2. Place the archive partition in the 'detached' section of your storage and unpack the archive:

   ```
   sudo unzip detached.zip -d <path to the 'detached' directory>
   ```

   For example:

   ```
   sudo unzip detached.zip -d /opt/kaspersky/kuma/clickhouse/data/store/d5b/d5bdd8d8-e1eb-4968-95bd-d8d8e1eb3968/detached/
   ```

3. Run the command to attach the partition:

   ```
   sudo /opt/kaspersky/kuma/clickhouse/bin/client.sh -d kuma --multiline --query "ALTER TABLE events_local_v2 ATTACH PARTITION ID '<partition ID>'"
   ```

   Repeat the steps of unpacking the archive and attaching the partition on each replica of the shard.

As a result, the archived partition is attached and its events are again available for search.

Creating a resource set for a storage

In the KUMA web interface, a storage service is created based on the resource set for the storage.

To create a resource set for a storage in the KUMA web interface:

1. In the KUMA web interface, in the **Resources** → **Storages** section, click **Create new**.

   This opens the **Create storage** window.

2. On the **Basic settings** tab, in the **Storage name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

3. In the **Tenant** drop-down list, select the tenant that will own the storage.

4. In the **Tags** drop-down list, select the tags for the resource set that you are creating.

   The list includes all available tags created in the tenant of the resource and in the Shared tenant. You can find a tag in the list by typing its name in the field. If the tag you entered does not exist, you can press **`Enter`** or click **Add** to create it.

5. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

6. In the **Retention policy options** field, select an event storage condition in the ClickHouse cluster for the storage, which, when satisfied, will cause events to be transferred to cold storage disks or deleted if cold storage is not configured or is configured incorrectly. The condition is applied to the default space and to events from deleted spaces.

   By default, ClickHouse moves events to cold storage disks or deletes them if more than 97% of the storage is full. KUMA also applies an additional **365 days** storage condition when creating a storage. You can configure custom storage conditions for more stable performance of the storage.

   To set the storage condition, do one of the following:

   - If you want to limit the storage period for events, select **Days** from the drop-down list, and in the field, specify the maximum event storage period (in days) in the ClickHouse hot storage cluster.

     After the specified period, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value is 1. The default value is 365.

     For [OOTV] Storage, the default event storage period is two days. If you want to use this storage, you can change the event storage condition for it, if necessary.

   - If you want to limit the maximum storage size, select **GB** from the drop-down list, and in the field, specify the maximum storage size in gigabytes.

     When the storage reaches the specified size, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum and default value is 1.

   - If you want to limit the storage size to a percentage of disk space that is available to the storage, select **Percent** from the drop-down list and specify the maximum storage size as a percentage of the available disk space.

     > You can get information about available disk space from ClickHouse by sending the `'SELECT hostname(), name, total_space FROM system.disks'` query.

When the storage reaches the specified percentage of disk space available to it, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. Possible values: 1 to 95. The default value is 80. If you want to use percentages for all storage spaces, the sum total of percentages in the conditions of all spaces may not exceed 95, but we recommend specifying a limit of at most 90% for the entire storage or for individual spaces.

We do not recommend specifying a value less than 5% because this increases the likelihood of data loss in the storage.

> If the storage is organized as a cluster, the amount of occupied disk space is determined for each of the several servers separately. Stored data may be unevenly distributed among disks on the servers. KUMA shifts events to cold storage disks or deletes them when the maximum storage size is reached *on any individual server in the cluster*.

The algorithm for determining the size of occupied storage space

The size of occupied storage space is determined as follows:

1. Information about the state of KUMA disks and partitions is obtained from all available replicas.
2. If occupied space exceeds the configured values, KUMA takes action depending on the type of data storage: hot or cold.

   For hot storage, the following actions are performed for each available node and space:

   a. The total size of the partitions of this space on the replica is calculated.

   b. The maximum space size on the replica is calculated:

   - If an absolute value in GB is set for the storage size, the maximum space size on the replica is calculated as follows:

     ```
     <maximum space size> = <configured threshold> / <number of replicas in the cluster>
     ```

     For example, consider a cluster with three shards, each of which contains two replicas; a threshold of 6 GB is set, and the maximum size of data on a replica is 1 GB.

   - If a relative value is set for the storage size as a percentage, the maximum space on the replica is calculated as follows:

     ```
     <maximum space size> = <space size on the replica> / <size of the replica disk> * 100%
     ```

     > The size of the replica disk does not include the space reserved for `ext4`.

   If the specified threshold is exceeded, partitions on the host are deleted or moved to cold storage. Partitions are deleted on all shards.

   For cold storage, partitions are deleted for each available space in accordance with the event retention period (TTL) configured in KUMA.

7. If you want to use an additional storage condition, click **Add policy options** and specify an additional storage condition as described in step 6.

The maximum number of conditions is two, and you can combine only conditions the following types:

- Days and storage size in GB
- Days and storage size as a percentage

If you want to delete a storage condition, click the X icon next to this condition.

8. In the **Audit retention period** field, specify the period, in days, to store audit events. The minimum value and default value is `365`.

9. If cold storage is required, specify the event storage term:

   - **Event retention time** specifies the total KUMA event storage duration in days, counting from the moment when the event is received. When the specified period expires, events are automatically deleted from the cold storage disk. The default value is 0.

     The event retention time is calculated as the sum of the event retention time in the ClickHouse hot storage cluster until the condition specified in the **Retention policy options** setting is triggered, and the event retention time on the cold storage disk. After one of storage conditions is triggered, the data partition for the earliest date is moved to the cold storage disk, and there it remains until the event retention time in KUMA expires.

     Depending on the specified storage condition, the resulting retention time is as follows:

     - If you specified a storage condition in days, the **Event retention time** must be strictly greater than the number of days specified in the storage condition. You can calculate the cold retention period for events as the **Events storage time** minus the number of days specified in the **Retention policy options** setting.

       If you do not want to store events on the cold storage disk, you can specify the same number of days in the **Event retention time** field as in the storage condition.

     - If you specified the storage condition in terms of disk size (absolute or percentage), the minimum value of the **Event retention time** is 1. The cold storage duration for events is calculated as **Event retention time** minus the number of days from the receipt of the event to triggering of the condition and the disk partition filling up, but until the condition is triggered, calculating an exact duration is impossible. In this case, we recommend specifying a relatively large value for **Event retention time** to avoid events being deleted.

       If you do not want to store events on the cold storage disk, you can set **Event retention time** to 0.

   - **Audit cold retention period**—the number of days to store audit events. The minimum value is 0.

   The **Events storage time** and **Audit cold retention period** settings become available only after at least one cold storage disk has been added.

10. If you want to change ClickHouse settings, in the **ClickHouse configuration override** field, paste the lines with settings from the ClickHouse configuration XML file /opt/kaspersky/kuma/clickhouse/cfg/config.xml. Specifying the root elements <yandex>, </yandex> is not required. Settings passed in this field are used instead of the default settings.

    Example:

    <merge_tree>

    <parts_to_delay_insert>600</parts_to_delay_insert>

    <parts_to_throw_insert>1100</parts_to_throw_insert>

    </merge_tree>

11. Use the **Debug** toggle switch to specify whether resource logging must be enabled. If you want to only log errors for all KUMA components, disable debugging. If you want to get detailed information in the logs, enable debugging.

12. If necessary, in the **ClickHouse cluster nodes** section, add ClickHouse cluster nodes to the storage.

    There can be multiple nodes. You can add nodes by clicking the **Add node** button or remove nodes by clicking the X icon of the relevant node.

    Available settings:

    - In the **FQDN** field, enter the fully qualified domain name of the node that you want to add. For example, `kuma-storage-cluster1-server1.example.com`.

    - In the **Shard ID**, **Replica ID**, and **Keeper ID** fields, specify the role of the node in the ClickHouse cluster. The shard and keeper IDs must be unique within the cluster, the replica ID must be unique within the shard. The following example shows how to populate the **ClickHouse cluster nodes** section for a storage with dedicated keepers in a **distributed installation**. You can adapt this example according to your needs.

      distributed

      Distributed Installation diagram

      Example:
      ClickHouse cluster nodes
      FQDN: kuma-storage-cluster1-server1.example.com
      Shard ID: 0
      Replica ID: 0
      Keeper ID: 1
      FQDN: kuma-storage-cluster1server2.example.com
      Shard ID: 0
      Replica ID: 0
      Keeper ID: 2
      FQDN: kuma-storage-cluster1server3.example.com
      Shard ID: 0
      Replica ID: 0
      Keeper ID: 3
      FQDN: kuma-storage-cluster1server4.example.com
      Shard ID: 1
      Replica ID: 1
      Keeper ID: 0
      FQDN: kuma-storage-cluster1server5.example.com
      Shard ID: 1
      Replica ID: 2
      Keeper ID: 0
      FQDN: kuma-storage-cluster1server6.example.com
      Shard ID: 2
      Replica ID: 1
      Keeper ID: 0
      FQDN: kuma-storage-cluster1server7.example.com
      Shard ID: 2
      Replica ID: 2
      Keeper ID: 0

13. If necessary, in the **Spaces** section, add spaces to the storage to distribute the stored events.

    There can be multiple spaces. You can add spaces by clicking the **Add space** button or remove spaces by clicking the X icon of the relevant space.

Available settings:

- In the **Name** field, specify a name for the space containing 1 to 128 Unicode characters.

- In the **Retention policy options** field, select an event storage condition in the ClickHouse cluster for the space, which, when satisfied, will cause events to be transferred to cold storage disks or deleted if cold storage is not configured or is configured incorrectly. KUMA applies the **365 days** storage condition when a space is added.

  To set the storage condition for a space, do one of the following:

  - If you want to limit the storage period for events, select **Days** from the drop-down list, and in the field, specify the maximum event storage period (in days) in the ClickHouse hot storage cluster.

    After the specified period, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value is 1. The default value is 365.

  - If you want to limit the maximum storage space size, select **GB** from the drop-down list, and in the field, specify the maximum space size in gigabytes.

    When the space reaches the specified size, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. The minimum value and default value is 1.

  - If you want to limit the space size to a percentage of disk space that is available to the storage, select **Percent** from the drop-down list, and in the field, specify the maximum space size as a percentage of the size of the disk available to the storage. In this case, the condition can also be triggered when the disk space available to the storage is decreased.

    When the space reaches the specified percentage of disk space available to the storage, events are automatically transferred to cold storage disks or deleted from the ClickHouse cluster, starting with the partitions with the oldest date. Possible values: 1 to 95. The default value is 80. If you want to use percentages for all storage spaces, the sum total of percentages in the conditions of all spaces may not exceed 95, but we recommend specifying a limit of at most 90% for the entire storage or for individual spaces.

    We do not recommend specifying small percentage values because this increases the probability of data loss in the storage.

  When using size as the storage condition, you must ensure that the total size of all spaces specified in the storage conditions does not exceed the physical size of the storage, otherwise an error will be displayed when starting the service.

  In storage conditions with a size limitation, use the same units of measure for all spaces of a storage (only gigabytes or only percentage values). Otherwise, if the condition is specified as a percentage for one space, and in gigabytes for another space, the storage may overflow due to mismatch of values, leading to data loss.

- If you want to make a space inactive if it is outdated and no longer relevant, select the **Read only** check box.

  This prevents events from going into that space. To make the space active again, clear the **Read only** check box. This check box is cleared by default.

- If necessary, in the **Event retention time** field, specify the total KUMA event storage duration in days, counting from the moment when the event is received. When the specified period expires, events are automatically deleted from the cold storage disk. The default value is 0.

The event retention time is calculated as the sum of the event retention time in the ClickHouse hot storage cluster until the condition specified in the **Retention policy options** setting is triggered, and the event retention time on the cold storage disk. After one of storage conditions is triggered, the data partition for the earliest date is moved to the cold storage disk, and there it remains until the event retention time in KUMA expires.

Depending on the specified storage condition, the resulting retention time is as follows:

- If you specified a storage condition in days, the **Event retention time** must be strictly greater than the number of days specified in the storage condition. The cold storage duration for events is calculated as the **Event storage time** minus the number of days specified in the **Retention policy options** setting.

  If you do not want to store events from this space on the cold storage disk, you can specify the same number of days in the **Event retention time** field as in the storage condition.

- If you specified the storage condition in terms of disk size (absolute or percentage), the minimum value of the **Event retention time** is 1. The cold storage duration for events is calculated as **Event retention time** minus the number of days from the receipt of the event to triggering of the condition and the disk partition filling up, but until the condition is triggered, calculating an exact duration is impossible. In this case, we recommend specifying a relatively large value for **Event retention time** to avoid events being deleted.

  If you do not want to store events from this space on the cold storage disk, you can set **Event retention time** to 0.

The **Event retention time** setting becomes available only after adding at least one cold storage disk.

- In the **Filter settings** section, you can specify conditions to identify events that will be put into this space. To create a new filter, in the **Filter** drop-down list, select an existing filter or **Create new**.

  Creating a filter in resources

  All fields of a KUMA event have a default value. For all numeric fields, the default value is 0. For all string fields, the default value is an empty string. An event not having a field means the field has the default value. Therefore, an event can trigger a filter even if the event does not have the field specified in the filter. For example, if you specify a `DestinationPort <= 1024` condition in the filter, but the event does not contain the `DestinationPort` field, the filter is triggered anyway because `DestinationPort = 0` by default.

  To create a filter:

  1. In the **Filter** drop-down list, select **Create new**.
  2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.
  3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.
  4. Under **Conditions**, specify the conditions that the events must meet:

     a. Click the **Add condition** button.

     b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

c. In the **Operator** drop-down list, select an operator.

Filter operators

- **=** means the left operand equals the right operand.
- **<** means the left operand is less than the right operand.
- **<=** means the left operand is less than or equal to the right operand.
- **>** means the left operand is greater than the right operand.
- **>=** means the left operand is greater than or equal to the right operand.
- **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains** means the left operand contains values of the right operand.
- **startsWith** means the left operand starts with one of the values of the right operand.
- **endsWith** means the left operand ends with one of the values of the right operand.
- **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable** checks for the presence of the entry in the specified context table.
- **intersect** checks for the presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, from the **If** drop-down list, select **If not**.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

After the service is created, you can view and delete spaces in the storage resource settings.

> There is no need to create a separate space for audit events. Events of this type (Type=4) are automatically placed in a separate Audit space with a storage term of at least 365 days. This space cannot be edited or deleted from the KUMA web interface.

14. If necessary, in the **Disks for cold storage** section, add to the storage the disks where you want to transfer events from the ClickHouse cluster for long-term storage.

    There can be multiple disks. You can add disks by clicking the **Add disk** button and remove them by clicking the **Remove disk** button.

    Available settings:

- In the **Type** drop-down list, select the type of the domain name of the disk being connected:
  - **Local**—for the disks mounted in the operating system as directories.
  - **HDFS**—for the disks of the Hadoop Distributed File System.
  - **S3**—for S3 cloud storage disks used in KUMA.

    > Starting with version 4.2, data stored on HDFS drives will be gradually migrated to S3 drives. Support for the HDFS file system will be discontinued in a future release of the application.

- In the **Name** field, specify the disk name. The name must contain 1 to 128 Unicode characters.

> You cannot change the name of an S3 drive without changing the path to the drive.

- If you select the **Local** domain name type for the disk, specify one of the following values in the **Path** field:
  - If the storage service is deployed locally, enter the absolute directory path of the mounted local disk. The path must begin and end withThe path must begin and end with a "/" character.

    The directory of the mounted local disk must be owned by the KUMA user.

  - If the storage service is deployed in the Kubernetes cluster, enter the path: /opt/cold. KUMA will use the /opt/cold directory to mount the /opt/lvp/cold directory of the Kubernetes cluster node where the storage service will be deployed.
- If you select **HDFS** domain name type for the disk, specify the path to HDFS in the **Host** field. For example, `hdfs://hdfs1:9000/clickhouse/`.
- If you selected **S3** as the domain name type for the disk, specify the following parameters:
  - In the **Host** field, enter the address of the S3 storage (for example, `https://www.company.com/clickhouse/storagefolder/`).

    Rules for specifying an S3 storage address

    When specifying an S3 storage address, the following requirements must be satisfied:

    - The address must be specified in the URL format, for example:

      `https://<web address/IP address of the server>/<path to disk>/`

    - The address must end with a "/" character.
    - The maximum length is 2048 characters.
    - Non-Latin letters are not allowed.

    You cannot change the address of an S3 disk without changing the name of the disk.

  - In the **Secret** field, specify the secret for access to the S3 storage. You can select a secret from the drop-down list or create a secret by clicking **Create new**.

15. In the **Configure backup and restore** section, manage the following settings:

    a. In the **Storage backup** field, specify the path at which you want to save the backup file. For example, /mnt/data.

    Path requirements:

    - The path must exist and be the same on all nodes of the storage cluster.
    - The user must have read, write, and execute permissions in the specified directory.
    - The specified directory must belong to the second or deeper nesting level. You cannot use a directory at the first nesting level, for example, /backup.
    - The directory name may not contain a period.

- **The path may not contain the following prefixes**.

  Prefixes that cannot be used when specifying paths to files:

  - `/*`
  - `/bin`
  - `/boot`
  - `/dev`
  - `/etc`
  - `/home`
  - `/lib`
  - `/lib64`
  - `/proc`
  - `/root`
  - `/run`
  - `/sys`
  - `/tmp`
  - `/usr/*`
  - `/usr/bin/`
  - `/usr/local/*`
  - `/usr/local/sbin/`
  - `/usr/local/bin/`
  - `/usr/sbin/`
  - `/usr/lib/`
  - `/usr/lib64/`
  - `/var/*`
  - `/var/lib/`
  - `/var/run/`
  - `/opt/kaspersky/kuma/`

These prefixes are used by the system as regular expressions for checking the path. The * character in prefixes matches any sequence of characters and must not be manually specified in the path itself if it is not part of a valid path.

Files are available at the following paths:

- `/opt/kaspersky/kuma/clickhouse/logs/`
- `/opt/kaspersky/kuma/metrics/log/`

Two directories are automatically created at the specified path: **backup** and **restore**.

If no path is specified, the scheduled backup, manual backup, restoration from backup, and import functions are not available.

b. If you want to configure scheduled backup, enable the **Scheduled backup** toggle switch.

c. In the **Back up after several days** field, specify an integer number of days after the creation of a partition after which you want a backup copy to be created. For example, if the value is **1**, backup is performed for the partition created the day before yesterday.

d. In the **Replicas** drop-down list, select one replica that you want to back up. Selecting multiple replicas of the same shard causes data duplication.

e. Under **Partitions**, click **Add partition** and manage the following settings:

a. In the **Tenant** drop-down list, select the tenant that contains the partition that you want to back up.

b. In the **Spaces** drop-down list, select one or more storage spaces that belong to the partition that you want to back up. If no value is specified, the whole partition is backed up.

You can manage settings for multiple partitions. If necessary, click **Add partition** and manage the settings of the new partition in the displayed fields. For example, if you want to back up the **Audit** space and you have 10 tenants, you must configure the backup separately for each tenant.

16. Go to the **Advanced settings** tab and fill in the following fields:

- In the **Buffer size** field, enter the buffer size in bytes at which events must be sent to the database. The default value is 128 MB. No maximum value is configured. If the virtual machine has less free RAM than the specified **Buffer size**, KUMA sets the limit to 128 MB.

  If the expected traffic to the storage service is greater than 1 Gbps, we recommend changing the **Buffer size** setting (in bytes) to the expected value and setting the **Buffer flush interval** to 2.

  The value of the **Buffer size** setting can be determined empirically. If the **Insert QPS** metric of the storage is greater than 1 and you see that a queue of requests building up, follow the recommendations for calculating and adjusting the **Buffer size** setting in the Insert QPS section provided in the Viewing KUMA metrics article.

- In the **Maximum buffer size** field, specify the maximum buffer size in bytes. The default value is 256 MB.

  By default, events are written to the database once per second. If the EPS rises and events are registered more often than once per second, the buffer size is automatically increased. The buffer is increased up to three times the **Buffer size** value, but cannot exceed the **Maximum buffer size** value. When the load goes back to normal, the buffer size stays the same; there is no automatic reduction of buffer size.

  If the **Maximum buffer size** is less than or equal to the **Buffer size**, the buffer size does not change.

  Restarting the storage service resets the increased buffer size to the value in the **Buffer size** field.

- In the **Buffer flush interval** field, enter the time in seconds for which KUMA waits for the buffer to fill up. If the buffer is not full, but the specified time has passed, KUMA sends events to the database. The default value is 1 second.

- In the **Required memory size** field, specify the minimum amount of free RAM in bytes. The default value is 128 MB.

  If the node on which the storage is installed has less free RAM than this setting, database insert request returns an error, and the events are recorded later. Limiting the memory size allows avoiding the situation when the memory fills up if, for any reason, the recording of events in the database is blocked and data begins to accumulate in memory.

- In the **Disk buffer size limit** field, enter a value in bytes. The disk buffer is used to temporarily store events that could not be sent for further processing or storage. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer. The default value is 10 GB.

- Use the **Disk buffer** toggle switch to enable or disable the disk buffer. By default, the disk buffer is enabled.

- Use the **Write to local database table** toggle switch to enable or disable writing to the local database table. Writing is disabled by default.

  If enabled, data is written only on the host on which the storage is located. We recommend using this functionality only if you have configured balancing on the collector and/or correlator at step **6. Routing**, in the **Advanced settings** section, the **URL selection policy** field is set to **Round robin**.

  If you disable writing, the data is distributed across the shards of the cluster.

- If necessary, use the **Debug** toggle switch to enable the logging of service operations.

- You can use the **Create dump periodically** toggle switch at the request of Technical Support to generate resource (CPU, RAM, etc.) utilization reports in the form of dumps.

- In the **Dump settings** field, you can specify the settings to be used when creating dumps. For instructions on filling out this field, please contact Technical Support.

The set of resources for the storage is created and is displayed under **Resources** → **Storages**. Now you can create a storage service.

Creating a storage service in the KUMA Console

When a set of resources is created for a storage, you can proceed to create a storage service in KUMA.

To create a storage service in the KUMA Console:

1. In the KUMA Console, under **Resources** → **Active services**, click **Add service**.

2. In the opened **Choose a service** window, select the set of resources that you just created for the storage and click **Create service**.

The storage service is created in the KUMA Console and is displayed under **Resources** → **Active services**. Now storage services must be installed to each node of the ClickHouse cluster by using the service ID.

Installing a storage in the KUMA network infrastructure

To create a storage:

1. Log in to the server where you want to install the service.

2. Create the /opt/kaspersky/kuma/ folder.

3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   > Make sure the kuma file has sufficient rights to run.

4. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma storage --core https://<KUMA Core server FQDN>:<port used by KUMA Core for internal communication (port 7210 by default)> --id <service ID copied from the KUMA console> --install
   ```

   Example: `sudo /opt/kaspersky/kuma/kuma storage --core https://kuma.example.com:7210 --id XXXXX --install`

   > When deploying several KUMA services on the same host, during the installation process you must specify unique ports for each component using the `--api.port < port >` parameter. The following setting values are used by default: `--api.port 7221`.

5. Repeat steps 1–2 for each storage node.

The storage is installed.

Backing up, restoring, and importing partitions

Users with the General Administrator role have access to the following KUMA actions:

- Scheduling partition backup.
- Manually backing up partitions.
- Restoring partitions from backup.
- Importing partitions from an external KUMA installation.

You can configure and use cold storage of events and partition backup at the same time.

You can view the status of partition backup, restoration, and import in an audit event. Also, after starting a manual backup of partitions, restoration of partitions from backup, or import of partitions from an external installation of KUMA, a corresponding task is created in the **Task manager** section.

Scheduling partition backup

Users with the General Administrator role can configure scheduled backup of partitions.

To schedule backup:

1. In the main menu, go to the **Monitoring** → **Resources and services** → **Active services** section and open an active storage to manage its settings.

2. In the **Edit storage** window that is opened, on the **Basic settings** tab, in the **Setting up backup and restoration** section, manage the following settings:

   a. In the **Storage backup** field, specify the path at which you want to save the backup file. For example, /mnt/data. If no path is specified, performing the backup is impossible.

   b. Enable the **Scheduled backup** toggle switch.

   c. In the **Backup after days** field, specify how many days after a partition is created you want a backup copy of it to be created. For example, if the value is **1**, backup is performed for the partition created the day before yesterday.

   d. In the **Replicas** drop-down list, select one replica that you want to back up. Selecting multiple replicas of the same shard causes data duplication.

3. Under **Backup partitions**, click **Add partition** and manage the following settings:

   a. In the **Tenant** drop-down list, select the tenant that contains the partition that you want to back up.

   b. In the **Spaces** drop-down list, select one or more storage spaces that belong to the partition that you want to back up. If no value is specified, the whole partition is backed up.

   You can manage settings for multiple partitions. If necessary, click **Add partition** and manage the settings of the new partition in the displayed fields. For example, if you want to back up the **Audit** space and you have 10 tenants, you must configure the backup separately for each tenant.

4. Click **Save**.

After saving the settings, the **backup** and **restore** directories are automatically created at the specified path. Subdirectories with tenant ID and space ID are automatically created in the **backup** directory. For example, if you specify /mnt/data as the backup path, the full path to the backup file is /mnt/data/backup/`<tenant ID>`/`<storage space ID>`/`<partition creation date>`.zip. If you need to look up the ID of a tenant or space, you can go to the **Events** section, open the event card and view the information that you need.

To restore from backup, you need to move the backup file to the **restore** directory.

Scheduled partition backup is configured. If a backup fails, a corresponding audit event is generated, the storage service gets the yellow status in the **Active services** section, and an error message is displayed when you hover over the yellow status indicator. The next time a scheduled backup is started, an attempt is made to back up the old data first.

Manually backing up partitions

Users with the General Administrator role can manually back up partitions.

To manually create a backup copy of a partition:

1. In the main menu, go to the **Monitoring** → **Resources and services** → **Active services** section and open an active storage to manage its settings.

2. This opens the **Edit storage** window; in that window, on the **Basic settings** tab, in the **Backup settings** section, in the **Storage backup** field, specify the path for the backup file. For example, /mnt/data. Make sure that the 'kuma' user has access rights to the specified directory.

3. In the main menu, go to the **Monitoring** → **Resources and services** → **Active services** section and select the check box next to the storage service with the green status.

4. In the toolbar or in the menu (⋮), select **Backup** and click **Back up**.

   The **Back up** option is available if one storage service is selected and it has the green status.

   This option is not available under the following conditions:

   - If at the previous step, you selected the check box next to more than one service.
   - If the storage service has the red status.
   - If no check boxes are selected.
   - If you have selected a check box next to a service of another type.

5. In the **Partition backup** window that is opened, manage the following settings:

   a. In the **Tenant** drop-down list, select the tenant that contains the partition that you want to back up.

   b. In the **Spaces** drop-down list, select one or more storage spaces that belong to the partition that you want to back up. If no value is specified, the whole partition is backed up.

   c. In the **Replicas** drop-down list, select one replica or all replicas that you want to back up. Selecting multiple replicas of the same shard causes data duplication.

   d. In the **Range of dates** field, use the calendar to specify the date range for the partitions to be included in the backup.

6. Click **OK**.

7. In the **Backup will be performed** confirmation window, click **OK**.

   After confirming the settings in the **Task manager** section, a partition backup task and an audit event are created.

Manual backup is completed.

After saving the settings, the **backup** and **restore** directories are automatically created at the specified path. Subdirectories with tenant ID and space ID are automatically created in the **backup** directory. For example, if you specify /mnt/data as the backup path, the full path to the backup file is /mnt/data/backup/`<tenant ID>`/`<storage space ID>`/`<partition creation date>`.zip. If you need to look up the ID of a tenant or space, you can go to the **Events** section, open the event card and view the information that you need.

To restore from backup, you need to move the backup file to the **restore** directory.

If the file name for manual backup is the same as the name of the scheduled backup file, an error is returned stating that the backup file cannot be saved. You must rename the file or delete the file and perform the manual backup again.

You can find out about the backup error in the **Active services** section by looking at the yellow status of the storage service and the error message, in the **Task manager** section, or the audit event.

Importing partitions from an external KUMA instance

Users with the General Administrator role can import partitions from an external KUMA installation.

To import partitions from an external KUMA installation:

1. Grant the 'kuma' user access rights to the backup directory that you specified when managing the backup settings, as well as access rights to the backup file itself using the following commands:

   ```
   sudo chown kuma <full path to the backup directory>
   sudo chown kuma *.zip
   ```

2. Place the file from which you want to import partitions into the **restore** directory. Partitions will be imported from this directory.

3. In the main menu, go to the **Monitoring** → **Resources and services** → **Active services** section, and select the check box next to the storage service with the green status.

4. In the toolbar or the menu (⋮), select **Backup** → **Import**.

5. In the **Import from an external installation of KUMA** window that is displayed, manage the following settings:

   a. In the **Tenant** drop-down list, select the tenant for which partitions will be imported.

   b. In the **Replicas** drop-down list, select one replica to which the partitions will be imported.

   c. In the **Fields for exclusions** drop-down list, specify the fields that you want to remove during import by selecting or clearing the check boxes.

      By default, the check boxes are selected for the following fields:

      - AggregationRuleName
      - CorrelationRuleName
      - SourceAccountID
      - DestinationAccountID
      - SourceAssetID
      - DestinationAssetID
      - DeviceAssetID
      - KUMA extended event schema fields:
        - Extended event schema fields
        - KSC asset fields
        - KLCorrelationRulePriority = "KL_CorrelationRulePriority"

- KLSourceAssetDisplayName = "KL_SourceAssetDisplayName"
- KLDestinationAssetDisplayName = "KL_DestinationAssetDisplayName"
- KLDeviceAssetDisplayName = "KL_DeviceAssetDisplayName"
- KLSourceAccountDisplayName = "KL_SourceAccountDisplayName"
- KLDestinationAccountDisplayName = "KL_DestinationAccountDisplayName"
- KLDllHijackingResult = "KL_AI_DLLHijackingCheckResult"

  When you select the **KUMA extended event schema fields** category for removal, the whole group of fields is removed. You cannot remove individual fields.

By default, the following fields are removed and therefore not available for selection in the web interface:

- ServiceID
- ServiceName
- CorrelationRuleID
- AggregationRuleID

6. Click **OK**.
7. In the **Import will be performed** confirmation window, click **OK**.

   After confirming the settings in the **Task manager** section, a partition import task and an audit event are created.

The import of partitions from an external KUMA installation is completed.

Restoring partitions from backup

Only users with the General Administrator role can restore partitions from backup.

To restore a partition from backup:

1. Grant the 'kuma' user access rights to the backup directory that you specified when managing the backup settings, as well as access rights to the backup file itself using the following commands:

   ```
   sudo chown kuma <full path to the backup directory>
   sudo chown kuma *.zip
   ```

2. Move the backup file from the **backup** directory to the **restore** directory.

3. In the main menu, go to the **Monitoring** → **Resources and services** → **Active services** section and select the check box next to the storage service with the green status.

4. In the toolbar or the menu (⋮), select **Backup** → **Restore**.

5. This opens the **Restore from the current KUMA installation** window; in that window, in the **Replicas** drop-down list, select one replica in which you want to perform restoration from backup.

6. Click **OK**.

7. In the **Restoration will be performed** confirmation window, click **OK**.

   After confirming the settings in the **Task manager** section, a partition restoration task and an audit event are created.

The restoration of partitions from backup is completed.

Creating a correlator

A correlator consists of two parts: one part is created inside the KUMA Console, and the other part is installed on the network infrastructure server intended for processing events.

Actions in the KUMA Console

A correlator is created in the KUMA Console by using the Installation Wizard. This Wizard combines the necessary resources into a set of resources for the correlator. Upon completion of the Wizard, the service itself is automatically created based on this set of resources.

To create a correlator in the KUMA Console,

Start the Correlator Installation Wizard:

- In the KUMA Console, under **Resources**, click **Add correlator**.
- In the KUMA Console, under **Resources** → **Correlators**, click **Add correlator**.

As a result of completing the steps of the Wizard, a correlator service is created in the KUMA Console.

A resource set for a correlator includes the following resources:

- Correlation rules
- Enrichment rules (if required)
- Response rules (if required)
- Destinations (normally one for sending events to a storage)

These resources can be prepared in advance, or you can create them while the Installation Wizard is running.

Actions on the KUMA correlator server

If you are installing the correlator on a server that you intend to use for event processing, you need to run the command displayed at the last step of the Installation Wizard on the server. When installing, you must specify the ID automatically assigned to the service in the KUMA Console, as well as the port used for communication.

Testing the installation

After creating a correlator, it is recommended to make sure that it is working correctly.

Starting the Correlator Installation Wizard

To start the Correlator Installation Wizard:

- In the KUMA Console, under **Resources**, click **Add correlator**.
- In the KUMA Console, under **Resources** → **Correlators**, click **Add correlator**.

  Follow the instructions of the Wizard.

Aside from the first and last steps of the Wizard, the steps of the Wizard can be performed in any order. You can switch between steps by clicking the **Next** and **Previous** buttons, as well as by clicking the names of the steps in the left side of the window.

After the Wizard completes, a resource set for the correlator is created in the KUMA Console under **Resources** → **Correlators**, and a correlator service is added under **Resources** → **Active services**.

Step 1. General correlator settings

This is a required step of the Installation Wizard. At this step, you specify the main settings of the correlator: the correlator name and the tenant that will own it.

To define the main settings of the correlator:

- In the **Name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.
- In the **Tenant** drop-down list, select the tenant that will own the correlator. The tenant selection determines what resources will be available when the collector is created.

  If you return to this window from another subsequent step of the Installation Wizard and select another tenant, you will have to manually edit all the resources that you have added to the service. Only resources from the selected tenant and shared tenant can be added to the service.

- If required, specify the number of processes that the service can run concurrently in the **Workers** field. By default, the number of worker processes is the same as the number of vCPUs on the server where the service is installed.
- If necessary, use the **Debug** drop-down list to enable logging of service operations.
- You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

The main settings of the correlator are defined. Proceed to the next step of the Installation Wizard.

Step 2. Global variables

If tracking values in event fields, active lists, or dictionaries is not enough to cover some specific security scenarios, you can use global and local variables. You can use them to take various actions on the values received by the correlators by implementing complex logic for threat detection. Variables can be assigned a specific function and then queried from correlation rules as if they were ordinary event fields, with the triggered function result received in response.

To add a global variable in the correlator,

click the **Add variable** button and specify the following parameters:

- In the **Variable** window, enter the name of the variable.

  Variable naming requirements

  - Must be unique within the correlator.
  - Must contain 1 to 128 Unicode characters.
  - Must not begin with the character $.
  - Must be written in camelCase or CamelCase.

- In the **Value** window, enter the variable function.

  Description of variable functions.

The global variable is added. It can be queried from correlation rules by adding the $ character in front of the variable name. There can be multiple variables. Added variables can be edited or deleted by using the ✕ icon.

Proceed to the next step of the Installation Wizard.

Step 3. Correlation

This is an optional but recommended step of the Installation Wizard. On the **Correlation** tab of the Installation Wizard, select or create correlation rules. These resources define the sequences of events that indicate security-related incidents. When these sequences are detected, the correlator creates a correlation event and an alert.

If you have added global variables to the correlator, all added correlation rules can query them.

Correlation rules that are added to the set of resources for the correlator are displayed in the table with the following columns:

- **Correlation rules**—name of the correlation rule resource.
- **Type**—type of correlation rule: **standard**, **simple**, **operational**. The table can be filtered based on the values of this column by clicking the column header and selecting the relevant values.
- **Actions**—list of actions that will be performed by the correlator when the correlation rule is triggered. These actions are indicated in the correlation rule settings. The table can be filtered based on the values of this column by clicking the column header and selecting the relevant values.

  Available values:

  - **Output**—correlation events created by this correlation rule are transmitted to other correlator resources: enrichment, response rule, and then to other KUMA services.
  - **Edit active list**—the correlation rule changes the active lists.
  - **Loop to correlator**—the correlation event is sent to the same correlation rule for reprocessing.
  - **Categorization**—the correlation rule changes asset categories.
  - **Event enrichment**—the correlation rule is configured to enrich correlation events.
  - **Do not create alert**—when a correlation event is created as a result of a correlation rule triggering, no alert is created for that. If you do not want to create an alert when a correlation rule is triggered, but you still want to send a correlation event to the storage, select the **Output** and **No alert** check boxes. If you select only the **No alert** check box, a correlation event is not saved in the storage.
  - **Shared resource**—the correlation rule or the resources used in the correlation rule are located in a shared tenant.

You can use the **Search** field to search for a correlation rule. Added correlation rules can be removed from the set of resources by selecting the relevant rules and clicking **Delete**.

Selecting a correlation rule opens a window with its settings, which can be edited and then saved by clicking **Save**. If you click **Delete** in this window, the correlation rule is unlinked from the set of resources.

Click the **Move up** and **Move down** buttons to change the position of the selected correlation rules in the table. It affects their execution sequence when events are processed. By clicking the **Move operational to top** button, you can move correlation rules of the **operational** type to the beginning of the correlation rules list.

To link the existing correlation rules to the set of resources for the correlator:

1. Click **Link**.

   The resource selection window opens.

2. Select the relevant correlation rules and click **OK**.

   The correlation rules will be linked to the set of resources for the correlator and will be displayed in the rules table.

To create a new correlation rule in a set of resources for a correlator:

1. Click **Add**.

   The correlation rule creation window opens.

2. Specify the correlation rule settings and click **Save**.

   The correlation rule will be created and linked to the set of resources for the correlator. It is displayed in the correlation rules table and in the list of resources under **Resources** → **Correlation rules**.

Proceed to the next step of the Installation Wizard.

Step 4. Enrichment

This is an optional step of the Installation Wizard. On the **Enrichment** tab of the Installation Wizard, you can select or create enrichment rules and indicate which data from which sources you want to add to correlation events that the correlator creates. There can be more than one enrichment rule. You can add them by clicking the **Add** button and can remove them by clicking the × button.

To add an existing enrichment rule to a set of resources:

1. Click **Add**.

   This opens the enrichment rule settings block.

2. In the **Enrichment rule** drop-down list, select the relevant resource.

   The enrichment rule is added to the set of resources for the correlator.

To create a new enrichment rule in a set of resources:

1. Click **Add**.

   This opens the enrichment rule settings block.

2. In the **Enrichment rule** drop-down list, select **Create new**.

3. In the **Source kind** drop-down list, select the source of data for enrichment and define its corresponding settings:

   - **constant**

     This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- **dictionary**

This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the " | " character when the dictionary is queried.

Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **event**

This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

- In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
- In the **Source field** drop-down list, select the event field whose value will be written to the target field.

- In the **Conversion** settings block, you can create rules for modifying the original data before it is written to the KUMA event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

  Available conversions

  Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

  - **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
  - **lower**—is used to make all characters of the value lowercase
  - **upper**—is used to make all characters of the value uppercase
  - **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
  - **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.
  - **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
    - **Replace chars** specifies the sequence of characters to be replaced.
    - **With chars** is the character sequence to be used instead of the character sequence being replaced.
  - **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.
  - **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
  - **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
  - **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:
    - **Expression** is the RE2 regular expression whose results you want to replace.
    - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- `{{- range $index, $element := . SA.StringArrayOne -}}`
  `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString .SA.StringArray}}
```

- **dns**

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers**—maximum number of requests per one point in time. The default value is `1`.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the Open Single Management Platform Core server.
- **Cache TTL**—the lifetime of the values stored in the cache. The default value is `60`.
- **Cache disabled**—you can use this drop-down list to enable or disable caching. Caching is enabled by default.
- **The Recursion desired** is available starting with Open Single Management Platform 3.4.1. You can use this toggle switch to make an Open Single Management Platform collector send recursive queries to authoritative DNS servers for the purposes of enrichment. By default it is **Disabled**.

- **cybertrace**

This type of enrichment is deprecated, we recommend using cybertrace-http instead.

This type of enrichment is used to add information from CyberTrace data streams to event fields.

Available settings:

- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests. The default CyberTrace port is 9999.
- **Number of connections**—maximum number of connections to the CyberTrace server that can be simultaneously established by KUMA. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.

- **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000,000.

- **Mapping** (required)—this settings block contains the mapping table for mapping KUMA event fields to CyberTrace indicator types. The **KUMA field** column shows the names of KUMA event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

  Available types of CyberTrace indicators:

  - **ip**
  - **url**
  - **hash**

  In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the × button to remove a string.

- **timezone**

  This type of enrichment is used in collectors and correlators to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

  When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

  > Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

  When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

  By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector or correlator) that processes the event. If the server time is changed, the service must be restarted.

  Permissible time formats when enriching the DeviceTimeZone field

  When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

4. Use the **Debug** drop-down list to indicate whether or not to enable logging of service operations. Logging is disabled by default.

5. In the **Filter** section, you can specify conditions to identify events that will be processed using the enrichment rule. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

   To create a filter:

   1. In the **Filter** drop-down list, select **Create new**.

   2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

   3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

   4. In the **Conditions** settings block, specify the conditions that the events must meet:

      a. Click the **Add condition** button.

      b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

      c. In the **operator** drop-down list, select an operator.

         Filter operators

         - **=**—the left operand equals the right operand.
         - **<**—the left operand is less than the right operand.
         - **<=**—the left operand is less than or equal to the right operand.
         - **>**—the left operand is greater than the right operand.
         - **>=**—the left operand is greater than or equal to the right operand.
         - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
         - **contains**—the left operand contains values of the right operand.
         - **startsWith**—the left operand starts with one of the values of the right operand.
         - **endsWith**—the left operand ends with one of the values of the right operand.
         - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

The new enrichment rule was added to the set of resources for the correlator.

Proceed to the next step of the Installation Wizard.

Step 5. Response

This is an optional step of the Installation Wizard. On the **Response** tab of the Installation Wizard, you can select or create response rules and indicate which actions must be performed when the correlation rules are triggered. There can be multiple response rules. You can add them by clicking the **Add** button and can remove them by clicking the ✕ button.

To add an existing response rule to a set of resources:

1. Click **Add**.

   The response rule settings window opens.

2. In the **Response rule** drop-down list, select the relevant resource.

The response rule is added to the set of resources for the correlator.

To create a new response rule in a set of resources:

1. Click **Add**.

   The response rule settings window opens.

2. In the **Response rule** drop-down list, select **Create new**.

3. In the **Type** drop-down list, select the type of response rule and define its corresponding settings:

   - **KSC response**—response rules for automatically launching the tasks on Kaspersky Security Center assets. For example, you can configure automatic startup of a virus scan or database update.

     Tasks are automatically started when KUMA is integrated with Kaspersky Security Center. Tasks are run only on assets that were imported from Kaspersky Security Center.

     Response settings

     - **Open Single Management Platform task** (required)—name of the Open Single Management Platform task that you need to start. Tasks must be created beforehand, and their names must begin with "`KUMA `". For example, "`KUMA antivirus check`".

       Types of Open Single Management Platform tasks that can be started using KUMA:

       - Update
       - Virus scan

- **Event field** (required)—defines the event field of the asset for which the Open Single Management Platform task should be started. Possible values:
  - SourceAssetID
  - DestinationAssetID
  - DeviceAssetID

  To send requests to Open Single Management Platform, you must ensure that Open Single Management Platform is available over the UDP protocol.

- **Run script**—response rules for automatically running a script. For example, you can create a script containing commands to be executed on the KUMA server when selected events are detected.

  The script file is stored on the server where the correlator service using the response resource is installed: /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts.

  The `kuma` user of this server requires the permissions to run the script.

  Response settings

  - **Timeout**—the number of seconds the system will wait before running the script.
  - **Script name** (required)—the name of the script file.

    If the script Response resource is linked to the Correlator service, but the is no script file in the /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts folder, the service will not start.
  - **Script arguments**—parameters or event field values that must be passed to the script.

    If the script includes actions taken on files, you should specify the absolute path to these files.

    Parameters can be written with quotation marks (").

    Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field which value must be passed to the script.

    Example: `-n "\"usr\": {{.SourceUserName}}"`

- **KEDR response**—response rules for automatically creating prevention rules, starting network isolation, or starting the application on Kaspersky Endpoint Detection and Response and Kaspersky Security Center assets.

  Automatic response actions are carried out when KUMA is integrated with Kaspersky Endpoint Detection and Response.

Response settings

- **Event field** (required)—event field containing the asset for which the response actions are needed. Possible values:
  - SourceAssetID
  - DestinationAssetID
  - DeviceAssetID
- **Task type**—response action to be performed when data matching the filter is received. The following types of response actions are available:
  - Enable network isolation.

    When selecting this type of response, you need to define values for the following settings:

    - **Isolation timeout**—the number of hours during which the network isolation of an asset will be active. You can indicate from 1 to 9,999 hours.

      If necessary, you can **add an exclusion for network isolation**.

      To add an exclusion for network isolation:

      a. Click the **Add exclusion** button.

      b. Select the direction of network traffic that must not be blocked:

      - Inbound.
      - Outbound.
      - Inbound/Outbound.

      c. In the **Asset IP** field, enter the IP address of the asset whose network traffic must not be blocked.

      d. If you selected **Inbound** or **Outbound**, specify the connection ports in the **Remote ports** and **Local ports** fields.

      e. If you want to add more than one exclusion, click **Add exclusion** and repeat the steps to fill in the **Traffic direction**, **Asset IP**, **Remote ports** and **Local ports** fields.

      f. If you want to delete an exclusion, click the **Delete** button under the relevant exclusion.

      When adding exclusions to a network isolation rule, Kaspersky Endpoint Detection and Response may incorrectly display the port values in the rule details. This does not affect application performance. For more details on viewing a network isolation rule, please refer to the *Kaspersky Anti Targeted Attack Platform Help Guide*.

  - Disable network isolation.

- Add prevention rule.

  When selecting this type of response, you need to define values for the following settings:

  - **Event fields to extract hash from**—event fields from which Kaspersky Unified Monitoring and Analysis Platform extracts SHA256 or MD5 hashes of the files that must be prevented from starting.

    The selected event fields and the values selected in the **Event field** must be added to the inherited fields of the correlation rule.

  - **File hash #1**—SHA256 or MD5 hash of the file to be blocked.

  At least one of the above fields must be completed.

- Delete prevention rule.

- Run program.

  When selecting this type of response, you need to define values for the following settings:

  - **File path**—path to the file of the process that you want to start.

  - **Command line parameters**—parameters with which you want to start the file.

  - **Working directory**—directory in which the file is located at the time of startup.

  When a response rule is triggered for users with the General Administrator role, the **Run program** task will be displayed in the **Task manager** section of the program web interface. The **Created by** column of the task table for this task displays **Scheduled task**. You can view task completion results.

  All of the listed operations can be performed on assets that have Kaspersky Endpoint Agent for Windows. On assets that have Kaspersky Endpoint Agent for Linux, the program can only be started.

  At the software level, the capability to create prevention rules and network isolation rules for assets with Kaspersky Endpoint Agent for Linux is unlimited. Kaspersky Unified Monitoring and Analysis Platform and Kaspersky Endpoint Detection and Response do not notify about failing to apply these rules.

- **Response via KICS for Networks**—response rules for automatically starting tasks on KICS for Networks assets. For example, you can change the asset status in KICS for Networks.

  Tasks are automatically started when KUMA is integrated with KICS for Networks.

Response settings

- **Event field** (required)—event field containing the asset for which the response actions are needed. Possible values:
  - SourceAssetID
  - DestinationAssetID
  - DeviceAssetID
- **KICS/KATA task**—response action to be performed when data matching the filter is received. The following types of response actions are available:
  - **Change asset status to Authorized**.
  - **Change asset status to Unauthorized**.

  When a response rule is triggered, KUMA will send KICS/KATA an API request to change the status of the specified device to **Authorized** or **Unauthorized**.

- **Response via Active Directory**—response rules for changing the permissions of Active Directory users. For example, block a user.

  Tasks are started if integration with Active Directory is configured.

  Response settings

  - **Account ID source**—event field, source of the Active Directory account ID value. Possible values:
    - SourceAccountID
    - DestinationAccountID
  - **AD command**—command that is applied to the account when the response rule is triggered. Available values:
    - Add account to group
    - Remove account from group
    - Reset account password
    - Block account

- In the **Workers** field, specify the number of processes that the service can run simultaneously.

  By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.

  This field is optional.

1. In the **Filter** section, you can specify conditions to identify events that will be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select an operator.

   Filter operators

   - **=**—the left operand equals the right operand.
   - **<**—the left operand is less than the right operand.
   - **<=**—the left operand is less than or equal to the right operand.
   - **>**—the left operand is greater than the right operand.
   - **>=**—the left operand is greater than or equal to the right operand.
   - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
   - **contains**—the left operand contains values of the right operand.
   - **startsWith**—the left operand starts with one of the values of the right operand.
   - **endsWith**—the left operand ends with one of the values of the right operand.
   - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
   - **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

     The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

     If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

The new response rule was added to the set of resources for the correlator.

Proceed to the next step of the Installation Wizard.

Step 6. Routing

This is an optional step of the Installation Wizard. On the **Routing** tab of the Installation Wizard, you can select or create destinations with settings indicating the forwarding destination of events created by the correlator. Events from a correlator are usually redirected to storage so that they can be saved and later viewed if necessary. Events can be sent to other locations as needed. There can be more than one destination point.

To add an existing destination to a set of resources for a correlator:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. In the **Destination** drop-down list, select the necessary destination.

   The window name changes to **Edit destination**, and it displays the settings of the selected resource. The resource can be opened for editing in a new browser tab by clicking the button.

3. Click **Save**.

The selected destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

To add a new destination to a set of resources for a correlator:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. Specify the settings on the **Basic settings** tab:

   - In the **Destination** drop-down list, select **Create new**.
   - In the **Name** field, enter a unique name for the destination resource. The name must contain 1 to 128 Unicode characters.
   - Click the **Disabled** toggle button to specify whether events will be sent to this destination. By default, sending events is enabled.

- Select the **Type** for the destination resource:
  - Select **storage** if you want to configure forwarding of processed events to the storage.
  - Select **correlator** if you want to configure forwarding of processed events to a correlator.
  - Select **nats-jetstream**, **tcp**, **http**, **kafka**, or **file** if you want to configure sending events to other locations.
- Specify the **URL** to which events should be sent in the hostname:<API port> format.

  You can specify multiple destination addresses by clicking the **URL** button for all types except **nats-jetstream** and **file**.
- For the **nats-jetstream** and **kafka** types, use the **Topic** field to specify which topic the data should be written to. The topic must contain Unicode characters. The Kafka topic is limited to 255 characters.

3. If necessary, specify the settings on the **Advanced settings** tab. The available settings vary based on the selected destination resource type:
   - **Compression** is a drop-down list where you can enable Snappy compression. By default, compression is **disabled**.
   - **Proxy** is a drop-down list for proxy server selection.
   - The **Buffer size** field is used to set buffer size (in bytes) for the destination. The default value is 1 MB, and the maximum value is 64 MB.
   - **Timeout** field is used to set the timeout (in seconds) for another service or component response. The default value is `30`.
   - **Disk buffer size limit** field is used to specify the size of the disk buffer in bytes. The default size is 10 GB.
   - **Cluster ID** is the ID of the NATS cluster.
   - **TLS mode** is a drop-down list where you can specify the conditions for using TLS encryption:
     - **Disabled** (default)—do not use TLS encryption.
     - **Enabled**—encryption is enabled, but without verification.
     - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

     When using TLS, it is impossible to specify an IP address as a URL.

- **URL selection policy** is a drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified:

  - **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination.

  - **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again.

  - **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.

- **Delimiter** is used to specify the character delimiting the events. By default, `\n` is used.

- **Path**—the file path if the **file** destination type is selected.

- **Buffer flush interval**—this field is used to set the time interval (in seconds) at which the data is sent to the destination. The default value is `100`.

- **Workers**—this field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.

- You can set health checks using the **Health check path** and **Health check timeout** fields. You can also disable health checks by selecting the **Health Check Disabled** check box.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

- The **Disk buffer disabled** drop-down list is used to enable or disable the use of a disk buffer. By default, the disk buffer is disabled.

- In the **Filter** section, you can specify the conditions to define events that will be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

  To create a filter:

  1. In the **Filter** drop-down list, select **Create new**.

  2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

  3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select an operator.

   Filter operators

   - **=**—the left operand equals the right operand.
   - **<**—the left operand is less than the right operand.
   - **<=**—the left operand is less than or equal to the right operand.
   - **>**—the left operand is greater than the right operand.
   - **>=**—the left operand is greater than or equal to the right operand.
   - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
   - **contains**—the left operand contains values of the right operand.
   - **startsWith**—the left operand starts with one of the values of the right operand.
   - **endsWith**—the left operand ends with one of the values of the right operand.
   - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
   - **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

     The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

     If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
   - **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

     If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
   - **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
   - **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

4. Click **Save**.

The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Proceed to the next step of the Installation Wizard.

Step 7. Setup validation

This is the required, final step of the Installation Wizard. At this step, KUMA creates a service resource set, and the Services are created automatically based on this set:

- The set of resources for the correlator is displayed under **Resources** → **Correlators**. It can be used to create new correlator services. When this set of resources changes, all services that operate based on this set of resources will start using the new parameters after the services restart. To do so, you can click the **Save and restart services** and **Save and update service configurations** buttons.

  A set of resources can be modified, copied, moved from one folder to another, deleted, imported, and exported, like other resources.

- Services are displayed in **Resources** → **Active services**. The services created using the Installation Wizard perform functions inside the KUMA program. To communicate with external parts of the network infrastructure, you need to install similar external services on the servers and assets intended for them. For example, an external correlator service should be installed on a server intended to process events, external storage services should be installed on servers with a deployed ClickHouse service, and external agent services should be installed on Windows assets that must both receive and forward Windows events.

To finish the Installation Wizard:

1. Click **Create and save service**.

   The **Setup validation** tab of the Installation Wizard displays a table of services created based on the set of resources selected in the Installation Wizard. The lower part of the window shows examples of commands that you must use to install external equivalents of these services on their intended servers and assets.

   For example:

   /opt/kaspersky/kuma/kuma correlator --core https://kuma-example:<port used for communication with the KUMA Core> --id <service ID> --api.port <port used for communication with the service> --install

   > The "kuma" file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

   The port for communication with the KUMA Core, the service ID, and the port for communication with the service are added to the command automatically. You should also ensure the network connectivity of the KUMA system and open the ports used by its components if necessary.

2. Close the Wizard by clicking **Save**.

The correlator service is created in KUMA. Now the equivalent service must be installed on the server intended for processing events.

Installing a correlator in a KUMA network infrastructure

A correlator consists of two parts: one part is created inside the KUMA Console, and the other part is installed on the network infrastructure server intended for processing events. The second part of the correlator is installed in the network infrastructure.

To install a correlator:

1. Log in to the server where you want to install the service.

2. Create the /opt/kaspersky/kuma/ folder.

3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   > Make sure the kuma file has sufficient rights to run.

4. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma correlator --core https://<KUMA Core server FQDN>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <service ID copied from the KUMA Console> --api.port <port used for communication with the installed component> --install
   ```

   Example: `sudo /opt/kaspersky/kuma/kuma correlator --core https://kuma.example.com:7210 --id XXXX --api.port YYYY --install`

You can copy the correlator installation command at the last step of the Installation Wizard. It automatically specifies the address and port of the KUMA Core server, the identifier of the correlator to be installed, and the port that the correlator uses for communication. Before installation, ensure the network connectivity of KUMA components.

When deploying several KUMA services on the same host, during the installation process you must specify unique ports for each component using the `--api.port <port>` parameter. The following setting values are used by default: `--api.port 7221`.

The correlator is installed. You can use it to analyze events for threats.

Validating correlator installation

To verify that the correlator is ready to receive events:

1. In the KUMA Console, go to the **Resources** → **Active services** section.
2. Make sure that the correlator you installed has the green status.

If the events that are fed into the correlator contain events that meet the correlation rule filter conditions, the events tab will show events with the `DeviceVendor=Kaspersky` and `DeviceProduct=KUMA` parameters. The name of the triggered correlation rule will be displayed as the name of these correlation events.

If no correlation events are found

You can create a simpler version of your correlation rule to find possible errors. Use a **simple** correlation rule and a single **Output** action. It is recommended to create a filter to find events that are regularly received by KUMA.

When updating, adding, or removing a correlation rule, you must update configuration of the correlator.

When you finish testing your correlation rules, you must remove all testing and temporary correlation rules from KUMA and update configuration of the correlator.

Creating the correlator correlator-ng

The correlator-ng correlator is a KUMA component for correlating and processing events from various sources. It allows filtering, combining, and analyzing events using correlation rules, selectors, and expressions.

In the current version, the correlator can only be managed using the YAML configuration. After initially getting a configuration from the KUMA Core, you make further changes to the settings in files; you cannot use the KUMA web interface to manage this correlator.

> Correlator-ng is not a drop-in replacement for the standard model of the correlator.

The main differences are:

- New syntax for rules and selectors, compatible with SQL-like expressions and functions.
- Support for join selectors: events from different selectors are compared in parallel in the order in which they are received.
- The `anyMatch` attribute can be used to trigger a correlation rule if at least one selector has reached its threshold value.
- Extended set of functions for working with data types, arrays, strings, timestamps, and CyberTrace fields.
- Stricter typing of expressions and automatic casting of field and variable values to the required type for correct handling of operations.
- The regular expression syntax has been changed: correlator-ng uses PCRE-like syntax, whereas the standard correlator model uses RE2.

Limitations of correlator-ng

The current implementation of correlator-ng has a number of limitations:

- The correlator cannot automatically classify assets based on incoming events and does not support actions that are usually performed when a response rule is triggered.
- The correlator cannot automatically enrich events with additional data from external sources, such as DNS queries, scripts for calculations and checks, or the CyberTrace system.
- Destinations of the **storage**, **correlator**, and **internal** types are supported only if they do not have a disk buffer.
- The `S.KL_EventRoute` field is not added to the correlation event.
- The correlator cannot enrich a correlation event with the following additional fields: `SourceAssetKSCInstanceID`, `SourceAssetKSCHostID`, `DestinationAssetKSCInstanceID`, `DestinationAssetKSCHostID`, `DeviceAssetKSCInstanceID`, `DeviceAssetKSCHostIDKSC`, `S.KL_SourceAccountDisplayName`, `S.KL_DestinationAccountDisplayName`.
- Templates are partially supported: some templates may not be processed, and functions of the `to_string` type may not work in some cases.

- A limited number of metrics is available.
- The `debug` parameter does not change the logging level.
- Bucket content can only be exported in JSON format. Importing and searching the contents of buckets is not available.
- The collection and analysis of correlator statistics for performance optimization is not supported.
- Regular expressions do not support constructs such as `(?=...)` and `(?<=...)`.
- Links to active lists, context tables, dictionaries, and asset categories must use unique IDs instead of names.

Configuring the correlator correlator-ng

Correlator-ng is deployed as a separate KUMA service. Its settings are defined by YAML configuration files and command line options of the process. After starting, the service establishes a connection with the KUMA Core, receives the configuration of rules and buckets, and proceeds to processing events.

Correlator-ng can be managed using a set of YAML files. Each file is described below.

Correlator-ng configuration files

File	Description
config.yaml	Contains basic information about the service, including the name, description, and general settings of the correlator.
correlation-rules.yaml	Contains the configuration of correlation rules, including the types of rules: standard, simple, operational, and periodical.
selector-includes.yaml	Contains nested filters that are linked to correlation rules for reusability.
active-lists.yaml	Contains the configuration of active lists.
context-tables.yaml	Contains the configuration of context tables.
dictionaries.yaml	Contains static dictionaries for use in selectors and correlation rules.
global-variables.yaml	Contains global variables available in correlation rules and selectors.
destinations.yaml	Contains event destinations (`storage`, `correlator`, `internal`) with IDs and sending settings.

The files are loaded at correlator startup in the following order: first config.yaml, then files with rules and selectors, then files with active lists, context tables, dictionaries, and global variables. The destination file is loaded last to ensure that events are routed correctly.

Systemd-unit and starting the correlator

The systemd unit file is used to start the correlator as a system service; this file defines the settings for starting, stopping and restarting correlator-ng.

Example configuration file:

```
[Unit]
Description=KUMA Correlator NG
StartLimitIntervalSec=1
After=network.target

[Service]
Type=simple
```

```
Restart=always
RestartPreventExitStatus=99
TimeoutSec=0
RestartSec=5

Environment="RUST_BACKTRACE=full"

User=kuma
Group=kuma

ExecStart=/opt/kuma correlator --id <correlator_service_ID> --core https://<FQDN_of_KUMA_Core>:7210 --wd
/opt/correlator-ng --api.port <API port of the correlator>
LimitFSIZE=infinity
LimitCPU=infinity
LimitAS=infinity
LimitNOFILE=500000
LimitNPROC=64000
LimitMEMLOCK=infinity
TasksMax=infinity
TasksAccounting=false

[Install]
WantedBy=multi-user.target
```

Correlator-ng logging

The `--logger.filter` option sets the logging level of correlator-ng. In this option, you can specify the same logging levels as in the standard `RUST_LOG` variable: `trace`, `debug`, `info`, `warn`, `error`.

Example:

```
--logger.filter debug
```

You can additionally use the `RUST_BACKTRACE` environment variable. Setting it to `full` enables the output of extended error information.

Supported data types in correlator-ng

Correlator-ng has a stricter data model compared to the standard version of the correlator. This affects correlation rules, selectors, and event processing functions. Each field of an event, variable, or selector now has a defined type. Automatic type casting is used only for compatible types; in other cases, the `to_string`, `to_integer`, and `to_float` conversion functions must be used.

The following data types are supported:

Supported data types

Data type	Description
String	String
VecString	Array of strings
I64	64-bit integer
VecI64	Array of 64-bit integers
F64	Float
VecF64	Array of floats
Timestamp	UNIX Timestamp (in milliseconds)
VecTimestamp	Array of UNIX Timestamps (in milliseconds)

Data type	Description
Uuid	Unique ID in the UUIDv4 format
VecUuid	Array of UUIDv4 values
Ip	IPv4 or IPv6 address
VecIp	Array of IP addresses
IpNet	Subnet with a CIDR prefix, for example, `192.168.1.0/24`.
VecIpNet	Array of subnets
Bool	Boolean value (`true` or `false`)
VecBool	Array of boolean values.

Stronger typing improves the efficiency of the correlator and reduces errors when compiling correlation rules. However, all operations with arrays, strings and timestamps must now take into account the data type, otherwise runtime errors can occur.

Correlation rule selectors and expressions

In correlator-ng, selectors can do all that variables can, and can use functions for working with strings, arrays, and numbers (for example, `substring`, `tr`, `to_lower`). It is not necessary to create a separate variable to use functions; the selectors themselves can use these functions.

Selectors are described using the SQL syntax applicable to `WHERE` clauses.

Selectors and variables are configured using expressions for filtering and merging events. They support the same operations as variables, with the ability to construct sophisticated conditions. The following expression types are available.

> When writing expressions for comparing values that involve constants, the constant must go on the right. This lets the correlator apply various optimizations. If the constant is in the left part, no optimization is done.

Constants

Constants

Constant	Type
`1`	I64 (integer)
`3.14`	F64 (float)
`'myString'`	String

References to values

References to values

Reference	Description
`DeviceVendor`	Value of `DeviceVendor` event field
`S.MyCustomField`	Value of `S.MyCustomField` composite field
`$myVariable`	Value of `myVariable` variable

Logical and group operators

Logical and group operators

Expression	Description
`<expression> AND <expression>`	Logical AND operator
`<expression> OR <expression>`	Logical OR operator
`NOT <expression>`	Negation of the result of an expression
`IS [NOT] NULL`	Checking for the absence of a value (in correlator-ng, an unset value is `NULL`)
`()`	Grouping expressions

Comparison operators

Comparison operators

Expression	Description
`<expression> = <expression>`	Checks for equality of values
`<expression> > <expression>`	Compares values using the `>` operator
`<expression> >= <expression>`	Compares values using the `>=` operator
`<expression> < <expression>`	Compares values using the `<` operator
`<expression> <= <expression>`	Compares values using the `<=` operator
`Type BETWEEN 1 AND 3`	Checks if the value of the `Type` field is between 1 and 3 inclusive

Arithmetic operators

Arithmetic operators

Expression	Description
`<expression> + <expression>`	Addition
`<expression> - <expression>`	Subtraction
`<expression> * <expression>`	Multiplication
`<expression> / <expression>`	Division
`<expression> % <expression>`	Modulo division

Features

Features

Function	Description
`eq_fold (<left operand expression>, <right operand expression>)`	Calls the `eq_fold` function.

For details about the available functions, see the Data processing functions in correlator-ng section.

Attributes of the 'standard' correlation rule

The **standard** correlation rule in correlator-ng supports additional attributes that allow modifying the conditions of a Full Match and specify the matching of events between selectors. These attributes extend the standard operation model of the correlation rule bucket and are applied at the stage when events are analyzed that have already been placed in the bucket.

The current implementation of correlator-ng supports two attributes:

- `anyMatch`
- `joinSelector`

Both attributes are applicable only to correlation rules of the **standard** type and are used directly in the configuration of the rule itself.

The anyMatch attribute

In the standard model of the correlator, the correlation rule is considered fulfilled when the bucket reaches the Full Match state, that is, when all selectors have accumulated the number of events equal to their threshold values.

If `anyMatch: true` is specified in the configuration of the correlation rule, the bucket reaches the Full Match state when any of the selectors of the rule (except for selectors of the **recovery** type) reaches its threshold value. Thus, the rule is triggered by the first fulfilled condition, without waiting for other selectors to accumulate events. Each selector continues to independently accumulate events in accordance with its configuration, however, correlation finishes immediately after one of the selectors reaches its threshold.

The correlation event generated by such a rule additionally contains the `$KL_selectorName` local variable. The value of this variable is equal to the tag of the selector whose threshold triggered the rule. This allows identifying the source of the triggering and applying various further processing steps to the result.

> A rule with the `anyMatch` attribute does not support a trigger of the **On timeout** type.

The JoinSelector attribute

In the standard model of the correlator, the correlation rule is considered fulfilled when the bucket reaches the Full Match state, that is, when all selectors have accumulated the number of events equal to their threshold values.

If `anyMatch: true` is specified in the configuration of the correlation rule, the bucket reaches the Full Match state when any of the selectors of the rule (except for selectors of the **recovery** type) reaches its threshold value. Thus, the rule is triggered by the first fulfilled condition, without waiting for other selectors to accumulate events. Each selector continues to independently accumulate events in accordance with its configuration, however, correlation finishes immediately after one of the selectors reaches its threshold.

The correlation event generated by such a rule additionally contains the `$KL_selectorName` local variable. The value of this variable is equal to the tag of the selector whose threshold triggered the rule. This allows identifying the source of the triggering and applying various further processing steps to the result.

> A rule with the `anyMatch` attribute does not support a trigger of the **On timeout** type.

The JoinSelector attribute

The `joinSelector` attribute allows additionally checking the relatedness of events accumulated by different selectors within the same bucket before it reaches the Full Match state.

The value of the attribute is an SQL-like expression, `joinSelector: <expression>`

The expression specifies the condition for comparing the events of different selectors. It must be a binary operation or function that accepts as its operands only constants and expressions of the form `with_selector('<selector_tag>', <expression>)`.

The `with_selector` function indicates that the expression is evaluated against the events of a particular selector.

When parsing the configuration, the correlator automatically determines the minimum threshold value of the selector among the selectors participating in the expression. The comparison is performed only for the number of events that is equal to this minimum value.

For example:

The selector threshold (number of events) for selector A is 5, and for selector B, it is 9. In the expression `with_selector('A', DeviceHostName) = with_selector('B', DeviceHostName)`, only the first five events of each selector are compared.

Events are compared in parallel in the order of their arrival: the first event of selector A is compared with the first event of selector B; the second with the second, and so on.

The `joinSelector` attribute can also use functions. For example: `joinSelector: contains_fold(with_selector('A', $var_A), with_selector('B', DeviceCustomString1))`

This example uses the `contains_fold` function, which checks for the presence of the value of the `$var_A` correlator variable, calculated for selector A, in the `DeviceCustomString1` field of events of selector B, case-insensitive.

The following functions are not supported in expressions of the `joinSelector` attribute:

- `has_ti_detect`
- `has_ti_detect_with_category`
- `has_ti_detect_with_indicator`
- `has_ti_detect_with_category_and_indicator`

Data functions in correlator-ng

Functions in correlator-ng are used in selectors and variables for working with event data, type casting, and filtering.

General syntax:

- Required arguments are enclosed in `< >`, optional arguments are enclosed in `[ ]`.
- Three dots `...` stand for an indefinite number of arguments.
- The return type is denoted by `-> <data type>`. The letter `T` indicates that the type depends on the type of the arguments.
- The type of the arguments can be concrete (for example, `String`, `I64`) or `T`, which stands for any available data type.

Type casting functions

These functions cast a value to a certain type.

Type casting functions

Function	Description
`to_string(< expression >)`	Casts the value to string.
`to_integer(< expression >)`	Casts the value to integer (I64).
`to_float(< expression >)`	Casts the value to float (F64).
`to_uuid(< expression >)`	Casts the value to UUIDv4.
`to_ip(< expression >)`	Casts the value to IPv4 or IPv6 address.
`to_ip_net(< expression >)`	Casts the value to an IPNet subnet.
`to_bool(< expression >)`	Casts the value to boolean.
`to_timestamp(< expression >, '[ timezone ]')`	Casts the value to a timestamp with an optional timezone.

Functions for comparing values

Functions for comparing values

Function	Description
`eq_fold(< left_operand >, < right_operand >) -> Bool`	Checks if the left-side value is the same as the right-side value, ignoring case.
`eq_any(< left_operand >, < right_operand 1>, [...]) -> Bool`	Checks if one of the right-side values is the same as the left-side value.
`eq_any_fold(< left_operand >, < right_operand >, [...]) -> Bool`	Compares the left-side value with multiple right-side values, ignoring case.
`contains(< left_operand >, < right_operand >) -> Bool`	Checks if the left-side value contains the right-side value.
`contains_fold(< left_operand >, < right_operand >) -> Bool`	Checks if the left-side value contains the right-side value, ignoring case.

Function	Description
`contains_any(< left_operand >, < right_operand 1>, [...]) -> Bool`	Checks if the left-side value contains any of the right-side values.
`contains_any_fold(< left_operand >, < right_operand >, [...]) -> Bool`	Checks if the left-side value contains any of the right-side values, ignoring case.
`starts_with(< left_operand >, < right_operand >) -> Bool`	Checks if the right-side value is a prefix of the left-side value.
`starts_with_fold(< left_operand >, < right_operand >) -> Bool`	Checks the prefix, case-insensitive.
`starts_with_any(< left_operand >, < right_operand 1>, [...]) -> Bool`	Checks if one of the right-side values is a prefix of the left-side value.
`starts_with_any_fold(< left_operand >, < right_operand 1>, [...]) -> Bool`	Checks the prefix for one of the right-side values, case-insensitive.
`ends_with(< left_operand >, < right_operand >) -> Bool`	Checks if the right-side value is a suffix of the left-side value.
`ends_with_fold(< left_operand >, < right_operand >) -> Bool`	Checks the suffix, case-insensitive.
`ends_with_any(< left_operand >, < right_operand 1>, [...]) -> Bool`	Checks if one of the right-side values is a suffix of the left-side value.
`ends_with_any_fold(< left_operand >, < right_operand 1>, [...]) -> Bool`	Checks the suffix, case-insensitive.
`match(< left_operand >, < right_operand >) -> Bool`	Checks if the left-side value matches the regular expression on the right side.
`match_any(< left_operand >, '< right_operand 1>', '[ right_operand 2]', [...]) -> Bool`	Checks if the left-side value matches one of the regular expressions on the right side.

Functions for working with active lists, dictionaries, and context tables

Functions for working with active lists, dictionaries, and context tables

Function	Description
`in_active_list('< active_list_ID >', < key_element 1>, [...]) -> Bool`	Checks for the presence of a value in the active list by key.
`in_context_table('< context_table_ID >', '< key_field_name 1>', < key_field_value 1>, '[ key_field_name 2]', [ key_field_value 2], [...]) -> Bool`	Checks for the presence of a value in the context table by key.
`in_dictionary('< dictionary_ID >', < key_element 1>, [ key_element 2], ...) -> Bool`	Checks for the presence of a value in the dictionary by key.
`dict('< dictionary_ID >', < key 1>, [...]) -> String`	Returns a value from a dictionary by key.
`table_dict('< dictionary_ID >', < column_name >, < key 1>, [...]) -> String`	Returns a value from a table dictionary.
`active_list('< active_list_ID >', < field_name >, < key 1>, [...]) -> String`	Returns a value from an active list.
`context_table('< context_table_ID >', '< field_name >', '< key_field >', < value >) -> T`	Returns the value of a field from the context table.
`context_table_list_item('< context_table_ID >', '< field_name >', < index >, ...) -> T`	Returns an array element from a context table.
`match_active_list(...)`	Returns a value from an active list if the key matches a regular expression.
`match_context_table(...)`	Returns a value from a context table if a regular expression is matched.
`match_dictionary(...)`	Returns a value from a dictionary if a regular expression is matched.

Subnet functions

Subnet functions

Function	Description
`in_subnet(< left_operand >, < subnet >) -> Bool`	Checks if an IP address belongs to a subnet.
`in_subnet_any(< left_operand >, < subnet 1>, [...]) -> Bool`	Checks if an IP address belongs to one of the subnets.
`in_subnet_active_list(...)`	Checks if an IP address belongs to a subnet from an active list.
`in_subnet_context_table(...)`	Checks if an IP address belongs to a subnet from a context table.
`in_subnet_dictionary(...)`	Checks if an IP address belongs to a subnet from a dictionary.

String functions

String functions

Function	Description
`to_lower(< operand >) -> String`	Converts the string to lower case.
`to_upper(< operand >) -> String`	Converts the string to upper case.
`substring(< string >, < head_of_string >, [ tail_of_string ]) -> String`	Returns a substring.
`index_of(< find_what >, < find_where >) -> I64`	Returns the position of the first occurrence.
`last_index_of(< find_what >, < find_where >) -> I64`	Returns the position of the last occurrence.
`tr(< operand >) -> String`	Trims whitespace at the beginning and end of a string.
`replace(< string >, < substring >, < replacement >) -> String`	Replaces a substring.
`regexp_replace(< string >, '< regular_expression >', '< replacement >') -> String`	Replaces a substring using a regular expression.
`regexp_capture(< string >, '< regular_expression >') -> String`	Gets a substring matching a regular expression.
`str_join('< separator >', < string 1>, [...]) -> String`	Joins multiple strings.

Array functions

Array functions

Function	Description
`append(< left operand >, < right operand >, '[ separator ]') -> T`	Adds a value to the end of an array or string.
`prepend(< left operand >, < right operand >, '[ separator ]') -> T`	Adds a value to the beginning of an array or string.
`len(< operand >) -> I64`	Returns the length of a string or array.
`item(< array >, < index >) -> T`	Returns an element of an array by index.
`distinct_items(< array >) -> T`	Removes duplicate elements from an array.
`sort_items(< array >, '< sorting_order >') -> T`	Sorts an array.

Timestamp functions

Timestamp functions

Function	Description
`now() -> Timestamp`	Returns the current timestamp.
`year(<timestamp>) -> I64`	Returns the year.
`month(<timestamp>) -> I64`	Returns the month number.
`month_str(<timestamp>) -> String`	Returns the month in string form.
`weekday(<timestamp>) -> I64`	Returns the number of the day of the week.
`weekday_str(<timestamp>) -> String`	Returns the day of the week in string form.
`day(<timestamp>) -> I64`	Returns the day of the month.
`hour(<timestamp>) -> I64`	Returns the hour.
`minute(<timestamp>) -> I64`	Returns the minute.
`second(<timestamp>) -> I64`	Returns the second.
`millisecond(<timestamp>) -> I64`	Returns the millisecond.
`format_timestamp(<timestamp>, '< format >') -> String`	Converts the timestamp to a string.
`truncate_timestamp(<timestamp>, '< duration >') -> Timestamp`	Rounds the timestamp to an integer value.

Time functions

Time functions

Function	Description
`duration_days(<t1>, <t2>) -> I64`	Returns the difference in days.
`duration_hours(<t1>, <t2>) -> I64`	Returns the difference in hours.
`duration_minutes(<t1>, <t2>) -> I64`	Returns the difference in minutes.
`duration_seconds(<t1>, <t2>) -> I64`	Returns the difference in seconds.
`duration_milliseconds(<t1>, <t2>) -> I64`	Returns the difference in milliseconds.

Mathematical functions

Mathematical functions

Function	Description
`round(< operand >) -> I64`	Rounds the value to the nearest integer.
`ceil(< operand >) -> I64`	Rounds the value up.
`floor(< operand >) -> I64`	Rounds the value down.
`abs(< operand >) -> T`	Returns the absolute value of a number.
`pow(< operand >, '< power >') -> T`	Raises a number to a power.
`math_min(< array >) -> T`	Returns the minimum element of an array.
`math_max(< array >) -> T`	Returns the maximum element of an array
`math_avg(< array >) -> F64`	Calculates the average value of an array.

Data decoding and analysis functions

Data decoding and analysis functions

Function	Description
`decode_hex_integer(< string >) -> I64`	Converts a hex string to a number.
`decode_hex(< string >) -> String`	Decodes a hex string.
`decode_base64(< string >) -> String`	Decodes a Base64 string.
`decode_url(< string >) -> String`	Decodes a URL-encoded string.
`entropy(< string >) -> F64`	Calculates the Shannon entropy of a string.
`has_bit(< left_operand >, < position >) -> Bool`	Checks if the bit is set at the specified position.
`has_vulnerability(< asset_ID >, (<CVE>), [ severity ]) -> Bool`	Checks if an asset has a vulnerability.

Other functions

Other functions

Function	Description
`include('< predicate_ID >', '[ predicate_name ]')`	Uses predefined predicates from the selector-includes.yaml file.
`ti('< category >', < indicator >, '< field >')`	Retrieves a value from a specialized TI field (CyberTrace).
`with_selector(' correlation_rule_selector_tag ', < expression >)`	Can only be used in the `joinSelector` attribute of a correlation rule to distinguish between sets of events.
`template('< template >', [ argument 1], [ argument 2], ...) -> String`	Generates a string based on a template by interpolating the values of event fields or variables into it.
`conditional('< predicate >', <then>, <else>)`	Returns one of two values, depending on the result of the predicate
`has_ti_detect*()`	Checks for a CyberTrace triggering by category or indicator.
`has_ti_detect_with_category('< category >')`	Checks for a CyberTrace triggering by category.
`has_ti_detect_with_indicator(< indicator >)`	Checks for a CyberTrace triggering by indicator.
`has_ti_detect_with_category_and_indicator('< category >', < indicator >)`	Checks for a CyberTrace triggering by category and indicator.

Creating an event router

An event router is a service that allows you to receive streams of events from collectors and correlators and then distribute the events to specified destinations in accordance with the configured filters.

To have events from the collector sent to the event router, you must create an `eventRouter` destination resource with the address of the event router and link the resource to the collectors that you want to send events to the event router.

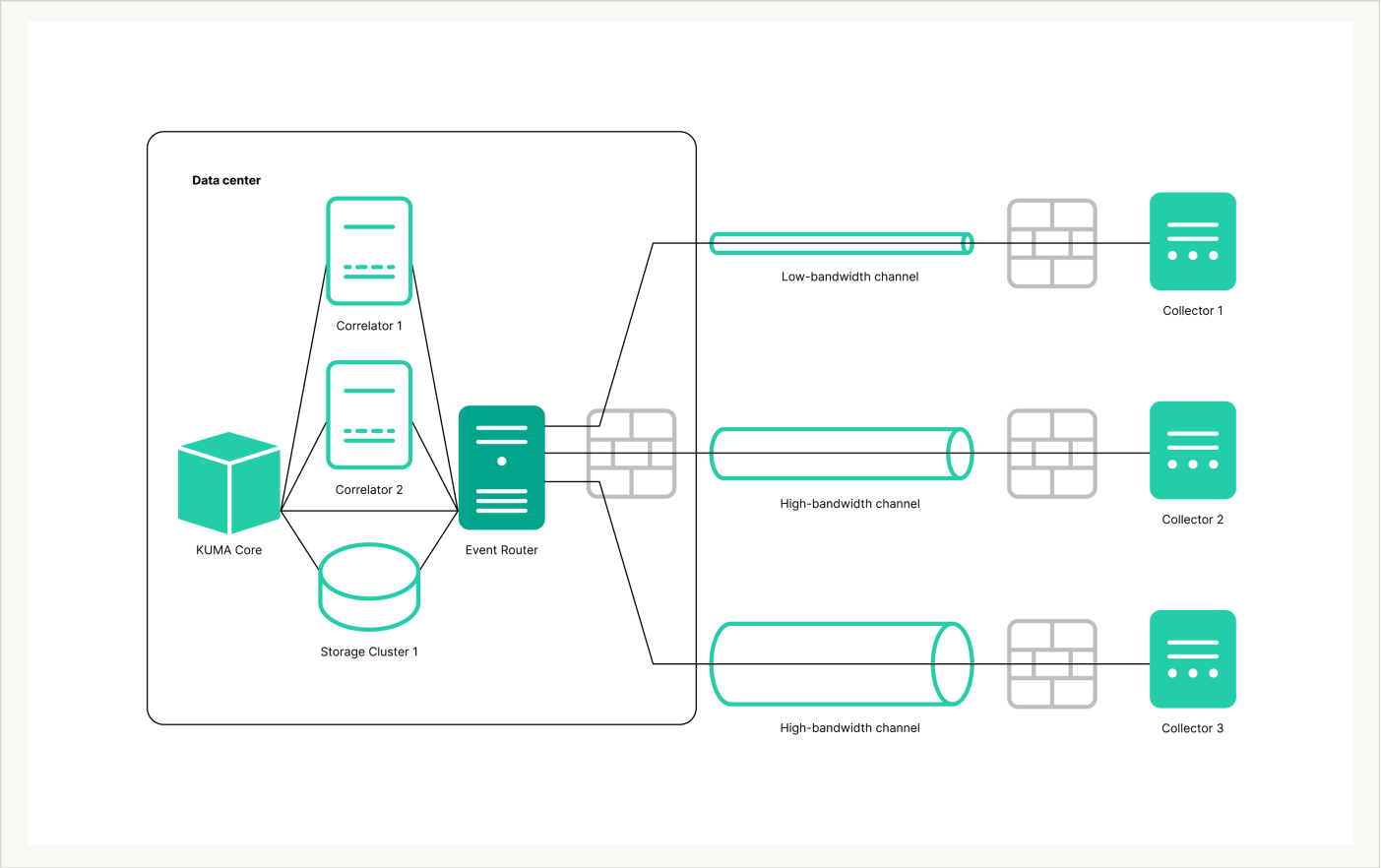
The event router receives events on the API port, just like `storage` and `correlator` destinations.

You can create a router in the **Resources** section.

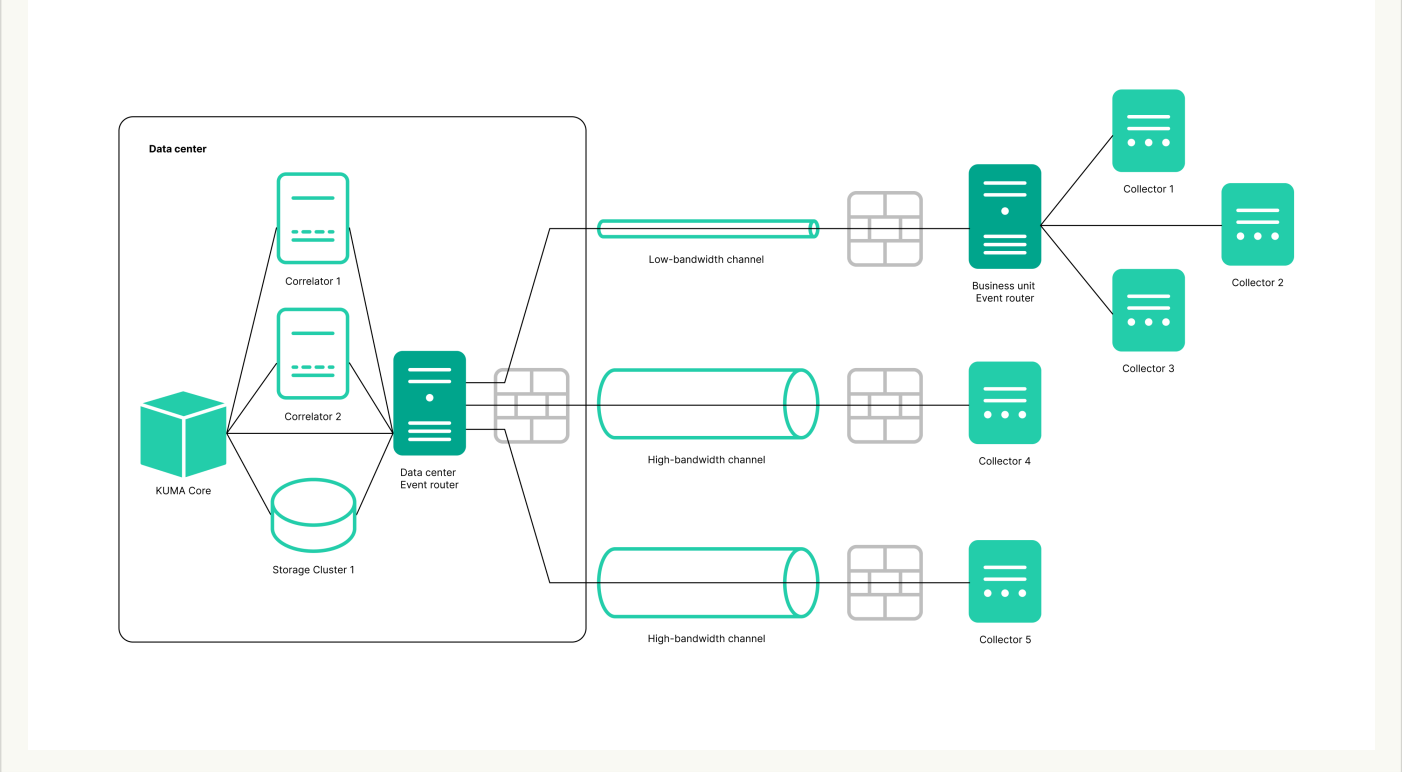
Using an event router lets you reduce the utilization of links, which is important for low-bandwidth and busy links.

Possible use cases:

Collector—Event router in the data center



Cascade connection: Multiple collectors—Event router at the branch office; Event router at the branch office —Event router in the data center



The event router must be installed on a Linux device. Only a user with the Main administrator role can create the service. You can create a service in any tenant; the tenant relation does not impose any restrictions.

You can use the following metrics to get information about the service performance:

- IO
- Process
- OS

As with other resources, the following audit events are generated for the event router in KUMA:

- Resource was successfully added
- Resource was successfully updated
- Resource was successfully deleted

Installing an event router involves two steps:

1. Create the event router service in the KUMA Console using the Installation Wizard.
2. Install the event router service on the server.

Starting the event router Installation Wizard

To start the event router Installation Wizard:

1. In the KUMA Console, in the **Resources** section, click **Event routers**.
2. In the **Event routers** window that opens, click **Add**.

Follow the instructions of the wizard.

Step 1. General settings of the event router

This is a required step of the Installation Wizard. At this step, you specify the main settings of the event router: its name and the tenant that will own it.

To specify the basic settings of the event router:

1. In the **Name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.
2. In the **Tenant** drop-down list, select the tenant that will own the event router. An event router belonging to a tenant is organizational in nature and does not impose any restrictions.
3. If necessary, specify the number of processes that the service can run concurrently in the **Handlers** field. By default, the number of handlers is the same as the number of vCPUs on the server where the service is installed.
4. If necessary, use the **Debug** toggle switch to enable logging of service operations.
5. You can optionally add up to 4000 Unicode characters describing the service in the **Description** field.

The basic settings of the event router are configured. Proceed to the next step of the Installation Wizard.

Step 2. Routing

This is a required step of the Installation Wizard. We recommend sending events to at least two destinations: to the correlator for analysis and to the storage for storage. You can also select another event router as the destination.

To specify the settings of the destination to which you want the event router to send events received from collectors:

1. In the **Routing** step of the Installation Wizard, click **Add**.

2. This opens the **Create destination** window; in that window, specify the following settings:

   a. On the **Basic settings** tab, in the **Name** field, enter a unique name for the destination. The name must contain 1 to 128 Unicode characters.

   b. You can use the **State** toggle switch to enable or disable the service as needed.

   c. In the **Type** drop-down list, select the type of the destination. The following values are available:

      - nats-jetstream
      - tcp
      - http
      - kafka
      - file
      - storage
      - correlator
      - eventRouter

   d. On the **Advanced settings** tab, specify the values of parameters. The set of parameters that can be configured depends on the type of the destination selected on the **Basic settings** tab. For detailed information about parameters and their values, click the link for each type of destination in paragraph "c." of this instruction.

The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Routing is configured. You can proceed to the next step of the Installation Wizard.

Step 3. Setup validation

This is the required, final step of the Installation Wizard.

To create an event router in the Installation Wizard:

1. Click **Create** and save service.

   The lower part of the window displays the command that you must use to install the event router on the server.

   Example command:

   ```
   /opt/kaspersky/kuma/kuma eventrouter --core https://kuma-example:<port used for communication with the KUMA Core> --id <event router service ID> --api.port <port used for communication with the service> --install
   ```

   The port for communication with the KUMA Core, the service ID, and the port for communication with the service are added to the command automatically. You must also ensure the network connectivity of KUMA and open the ports used by its components, if necessary.

2. Close the Wizard by clicking **Save**.

The service is installed in the KUMA Console. You can now proceed with installing the service in the KUMA network infrastructure.

Installing the event router on the server

To install the event router on the server:

1. Log in to the server where you want to install the event router service.

2. Create the /opt/kaspersky/kuma/ folder.

3. Copy the "kuma" file to the /opt/kaspersky/kuma/ directory. The file is located inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

4. Make sure the kuma file has sufficient rights to run. If the file is not executable, make it executable:

   ```
   sudo chmod +x /opt/kaspersky/kuma/kuma
   ```

5. Place the LICENSE file from the /kuma-ansible-installer/roles/kuma/files/ directory in the /opt/kaspersky/kuma/ directory and accept the license by running the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma license
   ```

6. Create the 'kuma' user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```

7. Make the 'kuma' user the owner of the /opt/kaspersky/kuma directory and all files inside the directory:

   ```
   sudo chown -R kuma:kuma /opt/kaspersky/kuma/
   ```

8. Add the KUMA event router port to firewall exclusions.

   > For the program to run correctly, ensure that the KUMA components are able to interact with other components and programs over the network via the protocols and ports specified during the installation of the KUMA components.

9. Execute the following command:

```
sudo /opt/kaspersky/kuma/kuma eventrouter --core https://<FQDN of the KUMA Core
server>:<port used by KUMA Core server for internal communication (port 7210 by
default)> --id <service ID copied from the KUMA Console> --api.port <port used for
communication with the installed component> --install
```

Example:

```
sudo /opt/kaspersky/kuma/kuma eventrouter --core https://kuma.example.com:7210 --id
XXXX --api.port YYYY --install
```

The event router is installed on the server. You can use it to receive events from collectors and relay the events to specified destinations.

Creating a collector

A collector consists of two parts: one part is created inside the KUMA Console, and the other part is installed on a server in the network infrastructure intended for receiving events.

Actions in the KUMA Console

A collector is created in the KUMA Console by using the Installation Wizard. This Wizard combines the necessary resources into a set of resources for the collector. Upon completion of the Wizard, the service itself is automatically created based on this set of resources.

To create a collector in the KUMA Console,

Start the Collector Installation Wizard:

- In the KUMA Console, in the **Resources** section, click **Add event source**.
- In the KUMA Console, in the **Resources** → **Collectors** section, click **Add collector**.

As a result of completing the steps of the Wizard, a collector service is created in the KUMA Console.

A resource set for a collector includes the following resources:

- Connector
- Normalizer (at least one)
- Filters (if required)
- Aggregation rules (if required)
- Enrichment rules (if required)
- Destinations (normally two are defined for sending events to the correlator and storage)

These resources can be prepared in advance, or you can create them while the Installation Wizard is running.

Actions on the KUMA Collector Server

When installing the collector on the server that you intend to use for receiving events, run the command displayed at the last step of the Installation Wizard. When installing, you must specify the ID automatically assigned to the service in the KUMA Console, as well as the port used for communication.

Testing the installation

After creating a collector, you are advised to make sure that it is working correctly.

Starting the Collector Installation Wizard

A collector consists of two parts: one part is created inside the KUMA Console, and the other part is installed on the network infrastructure server intended for receiving events. The Installation Wizard creates the first part of the collector.

To start the Collector Installation Wizard:

- In the KUMA Console, in the **Resources** section, click **Add event source**.
- In the KUMA Console, in the **Resources** → **Collectors** section, click **Add collector**.

Follow the instructions of the Wizard.

Aside from the first and last steps of the Wizard, the steps of the Wizard can be performed in any order. You can switch between steps by clicking the **Next** and **Previous** buttons, as well as by clicking the names of the steps in the left side of the window.

After the Wizard completes, a resource set for a collector is created in the KUMA Console under **Resources** → **Collectors**, and a collector service is added under **Resources** → **Active services**.

Step 1. Connect event sources

This is a required step of the Installation Wizard. At this step, you specify the main settings of the collector: its name and the tenant that will own it.

To specify the basic settings of the collector:

1. In the **Collector name** field, enter a unique name for the service you are creating. The name must contain 1 to 128 Unicode characters.

   When certain types of collectors are created, agents named "agent: <Collector name>, auto created" are also automatically created together with the collectors. If this type of agent was previously created and has not been deleted, it will be impossible to create a collector named <Collector name>. If this is the case, you will have to either specify a different name for the collector or delete the previously created agent.

2. In the **Tenant** drop-down list, select the tenant that will own the collector. The tenant selection determines what resources will be available when the collector is created.

   > If you return to this window from another subsequent step of the Installation Wizard and select another tenant, you will have to manually edit all the resources that you have added to the service. Only resources from the selected tenant and shared tenant can be added to the service.

3. If required, specify the number of processes that the service can run concurrently in the **Workers** field. By default, the number of worker processes is the same as the number of vCPUs on the server where the service is installed.

4. If necessary, use the **Debug** drop-down list to enable logging of service operations.

   Error messages of the collector service are logged even when debug mode is disabled. The log can be viewed on the machine where the collector is installed, in the /opt/kaspersky/kuma/collector/<collector ID>/log/collector directory.

5. You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

The main settings of the collector are specified. Proceed to the next step of the Installation Wizard.

Step 2. Transportation

This is a required step of the Installation Wizard. On the **Transport** tab of the Installation Wizard, select or create a connector and in its settings, specify the source of events for the collector service.

To add an existing connector to a resource set,

select the name of the required connector from the **Connector** drop-down list.

The **Transport** tab of the Installation Wizard displays the settings of the selected connector. You can open the selected connector for editing in a new browser tab by clicking the button.

To create a new connector:

1. Select **Create new** from the **Connector** drop-down list.

2. In the **Type** drop-down list, select the connector type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector:

   - tcp
   - udp
   - netflow
   - sflow
   - nats-jetstream
   - kafka
   - http

- sql
- file
- ftp
- nfs
- wmi
- wec
- snmp

When using the **tcp** or **udp** connector type at the normalization stage, IP addresses of the assets from which the events were received will be written in the DeviceAddress event field if it is empty.

When using a **wmi** or **wec** connector, agents will be automatically created for receiving Windows events.

It is recommended to use the default encoding (UTF-8), and to apply other settings only if bit characters are received in the fields of events.

> Making KUMA collectors to listen on ports up to 1,000 requires running the service of the relevant collector with root privileges. To do this, after installing the collector, add the line `AmbientCapabilities = CAP_NET_BIND_SERVICE` to its systemd configuration file in the [Service] section.
> The systemd file is located in the /usr/lib/systemd/system/kuma-collector-<`collector ID`>.service directory.

The connector is added to the resource set of the collector. The created connector is only available in this resource set and is not displayed in the web interface **Resources** → **Connectors section**.

Proceed to the next step of the Installation Wizard.

Step 3. Event parsing

This is a required step of the Installation Wizard. On the **Event parsing** tab of the Installation Wizard, select or create a normalizer whose settings will define the rules for converting raw events into normalized events. You can add multiple event parsing rules to the normalizer to implement complex event processing logic. You can test the normalizer using test events.

When creating a new normalizer in the Installation Wizard, by default it is saved in the set of resources for the collector and cannot be used in other collectors. The **Save normalizer** check box lets you create the normalizer as a separate resource, in which case the normalizer can be selected in other collectors of the tenant.

> If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Resources** → **Normalizers** in the web interface.

Adding a normalizer

To add an existing normalizer to a resource set:

1. Click the **Add event parsing** button.

   This opens the **Basic event parsing** window with the normalizer settings and the **Normalization scheme** tab active.

2. In the **Normalizer** drop-down list, select the required normalizer. The drop-down list includes normalizers belonging to the tenant of the collector and the Shared tenant.

   The **Basic event parsing** window displays the settings of the selected normalizer.

   If you want to edit the normalizer settings, in the **Normalizer** drop-down list, click the pencil icon next to the name of the relevant normalizer. This opens the **Edit normalizer** window with a dark circle. Clicking the dark circle opens the **Basic event parsing** window where you can edit the normalizer settings.

   If you want to edit advanced parsing settings, move the cursor over the dark circle to make a plus icon appear; click the plus icon to open the **Advanced event parsing** window. For details about configuring advanced event parsing, refer to the instruction below.

3. Click **OK**.

The normalizer is displayed as a dark circle on the **Basic event parsing** tab of the Installation Wizard. Clicking on the circle will open the normalizer options for viewing.

To create a new normalizer in the collector:

1. At the Event parsing step, on the **Parsing schemes** tab, click the **Add event parsing**.

   This opens the **Basic event parsing** window with the normalizer settings and the **Normalization scheme** tab active.

2. If you want to save the normalizer as a separate resource, select the **Save normalizer** check box; this makes the saved normalizer available for use in other collectors of the tenant. This check box is cleared by default.

3. In the **Name** field, enter a unique name for the normalizer. The name must contain 1 to 128 Unicode characters.

4. In the **Parsing method** drop-down list, select the type of events to receive. Depending on your choice, you can use the preconfigured rules for matching event fields or set your own rules. When you select some of the parsing methods, additional settings fields may need to be filled.

   Available parsing methods:

   - **json**

     This parsing method is used to process JSON data where each object, including its nested objects, occupies a single line in a file.

     When processing files with hierarchically structured data, you can reference the fields of nested objects using the dot notation. For example, the `username` parameter from the string `"user": {"username": "system: node: example-01"}` can be accessed by using the `user.username` query.

     Files are processed line by line. Multi-line objects with nested structures may be normalized incorrectly.

In complex normalization schemes where additional normalizers are used, all nested objects are processed at the first normalization level, except for cases when the extra normalization conditions are not specified and, therefore, the event being processed is passed to the extra normalizer in its entirety.

You can use `\n` and `\r\n` as newline characters. Strings must be UTF-8 encoded.

If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

- **cef**

  This parsing method is used to process CEF data.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**.

- **regexp**

  This parsing method is used to create custom rules for processing data in a format using regular expressions.

  You must add a regular expression (RE2 syntax) with named capturing groups to the field under **Normalization**. The name of the capturing group and its value are considered the field and value of the raw event that can be converted to an event field in KUMA format.

  To add event handling rules:

  1. If necessary, copy an example of the data you want to process to the **Event examples** field. We recommend completing this step.

  2. In the field under **Normalization**, add a RE2 regular expression with named capturing groups, for example, `"(?P<name>regexp)"`. The regular expression added to the field under **Normalization** must exactly match the event. When designing the regular expression, we recommend using special characters that match the starting and ending positions of the text: `^`, `$`.

     You can add multiple regular expressions or remove regular expressions. To add a regular expression, click **Add regular expression**. To remove a regular expression, click the delete icon ✕ next to it.

  3. Click the **Copy field names to the mapping table** button.

     Capture group names are displayed in the **KUMA field** column of the **Mapping** table. You can select the corresponding KUMA field in the column opposite each capturing group. If you followed the CEF format when naming the capturing groups, you can use automatic CEF mapping by selecting the **Use CEF syntax for normalization** check box.

  Event handling rules are added.

- **syslog**

  This parsing method is used to process data in syslog format.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**.

To parse events in rfc5424 format with a `structured-data` section, in the **Keep extra fields** drop-down list, select **Yes**. This makes the values from the `structured-data` section available in the `Extra` fields.

- **csv**

  This parsing method is used to create custom rules for processing CSV data.

  When choosing this parsing method, you must specify the separator of values in the string in the **Delimiter** field. Any single-byte ASCII character can be used as a delimiter for values in a string.

- **kv**

  This parsing method is used to process data in key-value pair format. Available parsing method settings are listed in the table below.

  Available parsing method settings

Setting	Description
Pair delimiter	The character used to separate key-value pairs. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Value delimiter** field.
Value delimiter	The character used to separate a key from its value. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Pair delimiter** field.

- **xml**

  This parsing method is used to process XML data in which each object, including nested objects, occupies a single line in a file. Files are processed line by line.

  If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

  If you select this parsing method, under **XML attributes**, you can specify the key XML attributes to be extracted from tags. If an XML structure has multiple XML attributes with different values in the same tag, you can identify the necessary value by specifying the key of the value in the **Source** column of the **Mapping** table.

  To add key XML attributes:

  1. Click **+ Add field**.

  2. This opens a window; in that window, specify the path to the XML attribute.

  You can add multiple XML attributes or remove XML attributes. To remove an individual XML attribute, click the delete icon ✕ next to it. To remove all XML attributes, click **Reset**.

  If XML key attributes are not specified, then in the course of field mapping the unique path to the XML value will be represented by a sequence of tags.

  ### Tag numbering

  Starting with KUMA 2.1.3, you can use automatic tag numbering in XML events. This lets you parse an event with the identical tags or unnamed tags, such as `<Data>`.

As an example, we will number the tags of the `EventData` attribute of the **Microsoft Windows PowerShell event ID 800**.

```
<Event xmlns="http://schemas .microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-ActiveDirectory_DomainService" Guid="
    {0e8478c5-3605-4e8c-8497-1e730c959516}" EventSourceName="NTDS" />
    <EventID Qualifiers="0000">0000</EventID>
    <Version>@</Version>
    <Level>4</Level>
    <Task>15</Task>
    <Opcode>0</Opcode>
    <Keywords >0x8080000000000000</Keywords>
    <TimeCreated SystemTime="2000-01-01T00:00:00.659495900Z" />
    <EventRecordID>55647</EventRecordID>
    <Correlation />
    <Execution ProcessID="1" ThreadID="1" />
    <Channel>service</Channel>
    <Computer>computer</Computer>
    <Security UserID="0000" />
  </System>
  <EventData>
    <Data>583</Data>
    <Data>36</Data>
    <Data>192.168.0.1:5084</Data>
    <Data>level</Data>
    <Data>name, lDAPDisplayName</Data>
    <Data />
    <Data>5545</Data>
    <Data>3</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>15</Data>
    <Data>none</Data>
  </EventData>
</Event>
```

To parse events with identical tags or unnamed tags, you need to configure tag numbering and data mapping for numbered tags with KUMA event fields.

KUMA 3.0.x supports using XML attributes and tag numbering at the same time in the same extra normalizer. If an XML attribute contains unnamed tags or identical tags, we recommend using tag numbering. If the XML attribute contains only named tags, we recommend using XML attributes.

To use XML attributes and tag numbering in extra normalizers, you must sequentially enable the **Keep raw event** setting in each extra normalizer along the path that the event follows to the target extra normalizer, and in the target extra normalizer itself.

For an example of how tag numbering works, you can refer to the **MicrosoftProducts** normalizer. The **Keep raw event** setting is enabled sequentially in both **AD FS** and **424** extra normalizers.

To set up the parsing of events with unnamed or identical tags:

1. Open an existing normalizer or create a new normalizer.

2. In the **Basic event parsing** window of the normalizer, in the **Parsing method** drop-down list, select **xml**.

3. In the **Tag numbering** field, click **+ Add field**.

4. In the displayed field, enter the full path to the tag to whose elements you want to assign a number, for example, `Event.EventData.Data`. The first tag gets number 0. If the tag is empty, for example, `<Data />`, it is also assigned a number.

5. To configure data mapping, under **Mapping**, click **+ Add row** and do the following:

   a. In the displayed row, in the **Source** field, enter the full path to the tag and the index of the tag. For example, for the Microsoft Windows PowerShell event ID 800 from the example above, the full paths to tags and tag indices are as follows:

      - `Event.EventData.Data.0,`
      - `Event.EventData.Data.1,`
      - `Event.EventData.Data.2` and so on.

   b. In the **KUMA field** drop-down list, select the field in the KUMA event that will receive the value from the numbered tag after parsing.

6. Save changes in one of the following ways:

   - If you created a new normalizer, click **Save**.
   - If you edited an existing normalizer, in the collector to which the normalizer is linked, click **Update configuration**.

Parsing is configured.

- **netflow5**

This parsing method is used to process data in the NetFlow v5 format.

If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **netflow5** parsing method is selected for the main parsing, extra normalization is not available.

> The default mapping rules for the **netflow5** parsing method do not specify the protocol type in KUMA event fields. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- **netflow9**

This parsing method is used to process data in the NetFlow v9 format.

If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **netflow9** parsing method is selected for the main parsing, extra normalization is not available.

The default mapping rules for the **netflow9** parsing method do not specify the protocol type in KUMA event fields. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- **sflow5**

  This parsing method is used to process data in sflow5 format.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **sflow5** parsing method is selected for the main parsing, extra normalization is not available.

- **ipfix**

  This parsing method is used to process IPFIX data.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **ipfix** parsing method is selected for the main parsing, extra normalization is not available.

  The default mapping rules for the **ipfix** parsing method do not specify the protocol type in KUMA event fields. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- **sql**—this method becomes available only when using a sql type connector.

  The normalizer uses this parsing method to process data obtained by making a selection from the database.

5. In the **Keep raw event** drop-down list, specify whether to store the original raw event in the newly created normalized event. Available values:

   - **Don't save**—do not save the raw event. This is the default setting.

   - **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. This value is convenient to use when debugging a service. In this case, every time an event has a non-empty `Raw` field, you know there was a problem.

     If fields containing the names `*Address` or `*Date*` do not comply with normalization rules, these fields are ignored. No normalization error occurs in this case, and the values of the fields are not displayed in the `Raw` field of the normalized event even if the **Keep raw event** → **Only errors** option was selected.

   - **Always**—always save the raw event in the `Raw` field of the normalized event.

6. In the **Keep extra fields** drop-down list, choose whether you want to store the raw event fields in the normalized event if no mapping rules have been configured for them (see below). The data is stored in the `Extra` event field. Normalized events can be searched and filtered based on the data stored in the `Extra` field.

   Filtering based on data from the Extra event field

   Conditions for filters based on data from the `Extra` event field:

   - Condition—**If**.
   - Left operand—**event field**.
   - In this event field, you can specify one of the following values:
     - `Extra` field.
     - Value from the `Extra` field in the following format:

       `Extra.<field name>`

       For example, `Extra.app`.

       You must specify the value manually.
     - Value from the array written to the `Extra` field in the following format:

       `Extra.<field name>.<array element>`

       For example, `Extra.array.0`.

       The values in the array are numbered starting from 0. You must specify the value manually. To work with a value in the `Extra` field at a depth of 3 and lower, you must use backticks ``, for example, `` `Extra.lev1.lev2.lev3` ``.
   - Operator – =.
   - Right operand—**constant**.
   - Value—the value by which you need to filter events.

   By default, fields are not saved.

7. Copy an example of the data you want to process to the **Event examples** field. This is an optional but recommended step.

8. In the **Mapping** table, configure the mapping of raw event fields to fields of the event in KUMA format:

   a. In the **Source** column, provide the name of the raw event field that you want to convert into the KUMA event field.

      For details about the field format, refer to the Normalized event data model article. For a description of the mapping, refer to the Mapping fields of predefined normalizers article.

      Clicking the 🔧 button next to the field names in the **Source** column opens the **Conversion** window, in which you can click the **Add conversion** button to create rules for modifying the original data before they are written to the KUMA event fields.

      Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:

  - **decodeHexString**—used to convert a HEX string to text.

  - **decodeBase64String**—used to convert a Base64 string to text.

  - **decodeBase64URLString**—used to convert a Base64url string to text.

When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

In the **Conversion** window, you can swap the added rules by dragging them by the ⠿ icon; you can also delete them using the × icon.

b. In the **KUMA field** column, select the required KUMA event field from the drop-down list. You can search for fields by entering their names in the field.

Recommendations concerning the KUMA field column

We recommend that you configure the mapping for the following KUMA fields. Otherwise, you will not be able to view observables in alert details and incident details.

The recommended KUMA fields depend on the observable types:

- For observables of the MD5 hash and SHA256 types:
  - FileHash
- For observables of the URL type:
  - RequestUrl
- For observables of the IP address type:
  - DeviceCustomIPv6Address1
  - DeviceCustomIPv6Address2
  - DeviceCustomIPv6Address3
  - DeviceCustomIPv6Address4

- DestinationTranslatedAddress
- DeviceTranslatedAddress
- DestinationAddress
- DeviceAddress
- SourceTranslatedAddress
- SourceAddress

- For observables of the Domain name type:
  - DestinationDnsDomain
  - DeviceDnsDomain
  - DeviceNtDomain
  - DestinationNtDomain
  - SourceDnsDomain
  - SourceNtDomain
- For observables of the UserName type:
  - DestinationUserName
  - SourceUserName
- For observables of the HostName type:
  - DestinationHostName
  - DeviceHostName
  - SourceHostName

c. If the name of the KUMA event field selected at the previous step begins with `DeviceCustom*` or `Flex*`, you can add a unique custom label in the **Label** field.

New table rows can be added by clicking the **Add row** button. Rows can be deleted individually by clicking the ✕ button or all at once by clicking the **Clear all** button.

If you want KUMA to enrich events with asset information, and the asset information to be available in the alert card when a correlation rule is triggered, in the **Mapping** table, configure a mapping of host address and host name fields depending on the purpose of the asset. For example, the mapping can apply to SourceAddress and SourceHostName, or DestinationAddress and DestinationHostName fields. As a result of enrichment, the event card includes a SourceAssetID or DestinationAssetID field, and a link to the asset card. Also, as a result of enrichment, asset information is available in the alert card.

If you have loaded data into the **Event examples** field, the table will have an **Examples** column containing examples of values carried over from the raw event field to the KUMA event field.

If the size of the KUMA event field is less than the length of the value placed in it, the value is truncated to the size of the event field.

9. Click **OK**.

The normalizer is displayed as a dark circle on the **Event parsing** tab of the Installation Wizard. If you want to open the normalizer settings for viewing, click the dark circle. When you hover the mouse over the circle, a plus sign is displayed. Click it to add event parsing rules (refer to the instruction below).

Enriching normalized events with additional data

You can add additional data to newly created normalized events by creating enrichment rules in the normalizer. These enrichment rules are stored in the normalizer where they were created. There can be more than one enrichment rule.

To add enrichment rules to the normalizer:

1. Select the main or additional normalization rule to open a window, and in that window, click the **Enrichment** tab.

2. Click the **Add enrichment** button.

   The enrichment rule parameter block appears. You can delete the group of settings by clicking the × button.

3. Select the enrichment type from the **Source kind** drop-down list. Depending on the selected type, there may be advanced settings that will also need to be completed.

   Available Enrichment rule source types:

   - **constant**

     This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

     Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

     If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

     If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

   - **dictionary**

     This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the " | " character when the dictionary is queried.

Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **table**

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, click the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by clicking the **Add new element** button. Columns can be deleted by clicking the ✕ button.

- **event**

This type of enrichment is used when you need to write a value from another event field to the current event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Target field	The KUMA event field that you want to populate with the data.
Source field	The event field whose value is written to the target field.

Clicking 🔧 opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the KUMA event fields. You can reorder and delete created rules. To change the position of a rule, click ⠿ next to it. To delete a rule, click ✕ next to it.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.
- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
  - **Replace chars** specifies the sequence of characters to be replaced.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.
- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:
  - **Expression** is the RE2 regular expression whose results you want to replace.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

When using enrichment of events that have **event** selected as the **Source kind** and the extended event schema fields are used as arguments, the following special considerations apply:

- If the source extended event schema field has the `Array of strings` type, and the target extended event schema field has the `String` type, the values are written to the target extended event schema field in TSV format.

  Example: The `SA.StringArray` extended event schema field contains values: `"string1"`, `"string2"`, `"string3"`. An event enrichment operation is performed. The result of the event enrichment operation is written to the `DeviceCustomString1` extended event schema field. The `DeviceCustomString1` extended event schema field contains values: `["string1", "string2", "string3"]`.

- If the source and target extended event schema fields have the `Array of strings` type, values of the source extended event schema field are added to the values of the target extended event schema field, and the `","` character is used as the delimiter character.

  Example: The `SA.StringArrayOne` field of the extended event scheme contains the `["string1", "string2", "string3"]` values, and the `SA.StringArrayTwo` field of the extended event scheme contains the `["string4", "string5", "string6"]` values. An event enrichment operation is performed. The result of the event enrichment operation is written to the `SA.StringArrayTwo` field of the extended event scheme. The `SA.StringArrayTwo` extended event schema field contains values: `["string4", "string5", "string6", "string1", "string2", "string3"]`.

- **template**

  This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

  If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

  - `{{.SA.StringArrayOne}}`
  - `{{- range $index, $element := . SA.StringArrayOne -}}`
    `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

  To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

  ```
  template {{toString .SA.StringArray}}
  ```

4. In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.

   This setting is not available for the enrichment source of the **Table** type.

5. If you want to enable details in the normalizer log, set the **Debug** toggle switch to enabled. Details are disabled by default.

6. Click **OK**.

Event enrichment rules with the additional data are added to the normalizer, to the selected parsing rule.

Configuring parsing linked to IP addresses

You can direct events from multiple IP addresses, from sources of different types, to the same collector, and the collector will apply the corresponding configured normalizers.

You can use this method for collectors with a connector of the UDP, TCP, or HTTP type. If a UDP, TCP, or HTTP connector is specified in the collector at the **Transport** step, then at the **Event parsing** step, you can specify multiple IP addresses on the **Parsing settings** tab and choose the normalizer that you want to use for events coming from the specified addresses. The following types of normalizers are available: json, cef, regexp, syslog, csv, kv, xml.

In a collector with configured normalizers linked to IP addresses, if you change the connector type to any type other than UDP, TCP, HTTP, the **Parsing settings** tab disappears and only the first of the previously specified normalizers is specified at the **Parsing** step. The tab disappears from the web interface immediately, but the changes are applied after the resource is saved. If you want to restore the previous settings, exit the collector Installation Wizard without saving.

For normalizers of the Syslog and regexp types, you can use a normalizer chain by specifying extra normalization conditions depending on the value of the DeviceProcessName field. The difference from extra normalization is that you can specify shared normalizers.

To configure parsing with linking to IP addresses:

1. At the **Event parsing** step, go to the **Parsing settings** tab.

2. In the **IP address(-es)** field, specify one or more IP addresses from which events will be received. You can specify multiple IP addresses separated by commas. Available format: IPv4. The length of the address list is unlimited; however, we recommend specifying a reasonable number of addresses to keep the load on the collector balanced. This field is mandatory if you want to apply multiple normalizers in one collector.

   Limitation: for each IP+normalizer combination, the IP address must be unique. KUMA checks the uniqueness of addresses, and if you specify the same IP address for different normalizers, the **The field must be unique** message is displayed.

   If you want to send all events to the same normalizer without specifying IP addresses, we recommend creating a separate collector. We also recommend creating a separate collector with one normalizer if you want to apply the same normalizer to events from a large number of IP addresses; this helps improve the performance.

3. In the **Normalizer** field, create a normalizer or select an existing normalizer from the drop-down list. The arrow next to the drop-down list takes you to the **Parsing schemes** tab.

   Normalization is triggered if you have a connector type configured: UDP, TCP, HTTP; the event source header must be specified in the HTTP case.

   Taking into account the available connectors, the following normalizer types are available for automatic source recognition: json, cef, regexp, syslog, csv, kv, xml.

4. If you selected the Syslog or regexp normalizer type, you can **Additional condition**. Conditional normalization is available if **Field mapping** for DeviceProcessName is configured in the main normalizer. Under **Condition**, specify the process name in the DeviceProcessName field and create a normalizer or select an existing normalizer from the drop-down list. You can specify multiple combinations of DeviceProcessName + normalizer, normalization is performed until the first match is achieved.

Parsing with linking to IP addresses is configured.

Creating a structure of event normalization rules

To implement a complex event processing logic, you can add multiple event parsing rules to the normalizer. Events are transmitted between the parsing rules depending on the specified conditions. The sequence of creating parsing rules is important. The event is processed sequentially, and its path is shown using arrows.

To create an additional parsing rule:

1. Create a normalizer (refer to the instruction above).

   The created normalizer is displayed in the window as a dark circle.

2. Hover the mouse over the circle and click the plus sign button that appears.

3. In the **Additional event parsing** window that opens, specify the parameters of the additional event parsing rule:

   - **Extra normalization conditions** tab:

     If you want to send a raw event for extra normalization, select **Yes** in the **Keep raw event** drop-down list. The default value is **No**. We recommend passing a raw event to normalizers of json and xml types. If you want to send a raw event for extra normalization to the second, third, etc nesting levels, at each nesting level, select **Yes** in the **Keep raw event** drop-down list.

     To send only the events with a specific field to the additional normalizer, specify this field in the **Field to pass into normalizer** field.

     On this tab, you can also define other conditions. When these conditions are met, the event is sent for additional parsing.

   - **Normalization scheme** tab:

     On this tab, you can configure event processing rules, similar to the main normalizer settings (refer to the instruction above). The **Keep raw event** setting is not available. The **Event examples** field displays the values specified when the initial normalizer was created.

   - **Enrichment** tab:

     On this tab, you can configure event enrichment rules (refer to the instruction above).

4. Click **OK**.

   The additional parsing rule is added to the normalizer and displayed as a dark block with the conditions under which this rule is triggered. You can change the settings of the additional parsing rule by clicking it. If you hover the mouse over the additional parsing rule, a plus button appears. You can click this button to create a new additional parsing rule. To delete a normalizer, click the button with the trash icon.

The upper right corner of the window contains a search window where you can search parsing rules by name.

Proceed to the next step of the Installation Wizard.

Step 4. Filtering events

This is an optional step of the Installation Wizard. The **Event filtering** tab of the Installation Wizard allows you to select or create a filter whose settings specify the conditions for selecting events. You can add multiple filters to the collector. You can swap the filters by dragging them by the ⣿ icon as well as delete them. Filters are combined by the AND operator.

To add an existing filter to a collector resource set,

Click the **Add filter** button and select the required filter from the **Filter** drop-down menu.

To add a new filter to the collector resource set:

1. Click the **Add filter** button and select **Create new** from the **Filter** drop-down menu.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. This can be useful if you decide to reuse the same filter across different services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter in the **Name** field. The name must contain 1 to 128 Unicode characters.

4. In the **Conditions** section, specify the conditions that must be met by the filtered events:

   - The **Add condition** button is clicked to add filtering conditions. You can select two values (two operands, left and right) and assign the operation you want to perform with the selected values. The result of the operation is either True or False.

     - In the **operator** drop-down list, select the function to be performed by the filter.

       In this drop-down list, you can select the **do not match case** check box if the operator should ignore the case of values. This check box is ignored if the **InSubnet**, **InActiveList**, **InCategory**, and **InActiveDirectoryGroup** operators are selected. This check box is cleared by default.

       Filter operators

       - **=**—the left operand equals the right operand.
       - **<**—the left operand is less than the right operand.
       - **<=**—the left operand is less than or equal to the right operand.
       - **>**—the left operand is greater than the right operand.
       - **>=**—the left operand is greater than or equal to the right operand.
       - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
       - **contains**—the left operand contains values of the right operand.
       - **startsWith**—the left operand starts with one of the values of the right operand.
       - **endsWith**—the left operand ends with one of the values of the right operand.
       - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
       - **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

         The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

         If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

- In the **Left operand** and **Right operand** drop-down lists, select where the data to be filtered will come from. As a result of the selection, Advanced settings will appear. Use them to determine the exact value that will be passed to the filter. For example, when choosing **active list** you will need to specify the name of the active list, the entry key, and the entry key field.

- You can use the **If** drop-down list to choose whether you need to create a negative filter condition.

Conditions can be deleted by clicking the × button.

- The **Add group** button is clicked to add groups of conditions. Operator **AND** can be switched between **AND**, **OR**, and **NOT** values.

  A condition group can be deleted by clicking the × button.

- By clicking **Add filter**, you can add existing filters selected in the **Select filter** drop-down list to the conditions. You can click ↗ to navigate to a nested filter.

  A nested filter can be deleted by clicking the × button.

The filter has been added.

Proceed to the next step of the Installation Wizard.

Step 5. Event aggregation

This is an optional step of the Installation Wizard. The **Event aggregation** tab of the Installation Wizard allows you to select or create an aggregation rule whose settings specify the conditions for aggregating events of the same type. You can add multiple aggregation rules to the collector.

To add an existing aggregation rule to a set of collector resources,

click **Add aggregation rule** and select **Aggregation rule** in the drop-down list.

To add a new aggregation rule to a set of collector resources:

1. Click the **Add aggregation rule** button and select **Create new** from the **Aggregation rule** drop-down menu.

2. Enter the name of the newly created aggregation rule in the **Name** field. The name must contain 1 to 128 Unicode characters.

3. In the **Threshold** field, specify how many events must be accumulated before the aggregation rule triggers and the events are aggregated. The default value is `100`.

4. In the **Triggered rule lifetime** field, specify how long (in seconds) the collector must accumulate events to be aggregated. When this time expires, the aggregation rule is triggered and a new aggregation event is created. The default value is `60`.

5. In the **Identical fields** section, click the **Add field** button to select the fields that will be used to identify the same types of events. Selected events can be deleted by clicking the buttons with a cross icon.

6. In the **Unique fields** section, you can click **Add field** to select the fields that will disqualify events from aggregation even if the events contain fields listed in the **Identical fields** section. Selected events can be deleted by clicking the buttons with a cross icon.

7. In the **Sum fields** section, you can click the **Add field** button to select the fields whose values will be summed during the aggregation process. Selected events can be deleted by clicking the buttons with a cross icon.

8. In the **Filter** section, you can specify the conditions to define events that will be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

   To create a filter:

   1. In the **Filter** drop-down list, select **Create new**.

   2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

   3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

   4. In the **Conditions** settings block, specify the conditions that the events must meet:

      a. Click the **Add condition** button.

      b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

      c. In the **operator** drop-down list, select an operator.

Filter operators

- **=**—the left operand equals the right operand.
- **<**—the left operand is less than the right operand.
- **<=**—the left operand is less than or equal to the right operand.
- **>**—the left operand is greater than the right operand.
- **>=**—the left operand is greater than or equal to the right operand.
- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Aggregation rule added. You can delete it by clicking the button.

Proceed to the next step of the Installation Wizard.

Step 6. Event enrichment

This is an optional step of the Installation Wizard. On the **Event enrichment** tab of the Installation Wizard, you can specify which data from which sources should be added to events processed by the collector. Events can be enriched with data obtained using enrichment rules or LDAP.

Rule-based enrichment

There can be more than one enrichment rule. You can add them by clicking the **Add enrichment** button and can remove them by clicking the button. You can use existing enrichment rules or create rules directly in the Installation Wizard.

To add an existing enrichment rule to a set of resources:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select the relevant resource.

The enrichment rule is added to the set of resources for the collector.

To create a new enrichment rule in a set of resources:

1. Click **Add enrichment**.

   This opens the enrichment rules settings block.

2. In the **Enrichment rule** drop-down list, select **Create new**.

3. In the **Source kind** drop-down list, select the source of data for enrichment and define its corresponding settings:

- **constant**

  This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

  If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

  If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- **dictionary**

  This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

  If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

  Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

  If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the " | " character when the dictionary is queried.

  Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **event**

  This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

  - In the **Target field** drop-down list, select the KUMA event field to which you want to write the data.
  - In the **Source field** drop-down list, select the event field whose value will be written to the target field.
  - In the **Conversion** settings block, you can create rules for modifying the original data before it is written to the KUMA event fields. The conversion type can be selected from the drop-down list. You can use the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

    Available conversions

    Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

    - **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
    - **lower**—is used to make all characters of the value lowercase
    - **upper**—is used to make all characters of the value uppercase
    - **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
    - **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.
    - **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
      - **Replace chars** specifies the sequence of characters to be replaced.
      - **With chars** is the character sequence to be used instead of the character sequence being replaced.
    - **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.
    - **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:

  - **decodeHexString**—used to convert a HEX string to text.

  - **decodeBase64String**—used to convert a Base64 string to text.

  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.

- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.

- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- `{{- range $index, $element := . SA.StringArrayOne -}}`
  `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString .SA.StringArray}}
```

- **dns**

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers**—maximum number of requests per one point in time. The default value is `1`.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the Open Single Management Platform Core server.
- **Cache TTL**—the lifetime of the values stored in the cache. The default value is `60`.
- **Cache disabled**—you can use this drop-down list to enable or disable caching. Caching is enabled by default.
- **The Recursion desired** is available starting with Open Single Management Platform 3.4.1. You can use this toggle switch to make an Open Single Management Platform collector send recursive queries to authoritative DNS servers for the purposes of enrichment. By default it is **Disabled**.

- **cybertrace**

This type of enrichment is deprecated, we recommend using cybertrace-http instead.

This type of enrichment is used to add information from CyberTrace data streams to event fields.

Available settings:

- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests. The default CyberTrace port is 9999.
- **Number of connections**—maximum number of connections to the CyberTrace server that can be simultaneously established by KUMA. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.
- **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000,000.
- **Mapping** (required)—this settings block contains the mapping table for mapping KUMA event fields to CyberTrace indicator types. The **KUMA field** column shows the names of KUMA event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

  Available types of CyberTrace indicators:

  - **ip**
  - **url**
  - **hash**

  In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the × button to remove a string.

- **timezone**

This type of enrichment is used in collectors and correlators to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

> Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector or correlator) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field

When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

- **geographic data**

This type of enrichment is used to add IP address geographic data to event fields. Learn more about linking IP addresses to geographic data.

When this type is selected, in the **Mapping geographic data to event fields** settings block, you must specify from which event field the IP address will be read, select the required attributes of geographic data, and define the event fields in which geographic data will be written:

1. In the **Event field with IP address** drop-down list, select the event field from which the IP address is read. Geographic data uploaded to KUMA is matched against this IP address.

   You can use the **Add event field with IP address** button to specify multiple event fields with IP addresses that require geographic data enrichment. You can delete event fields added in this way by clicking the **Delete event field with IP address** button.

   When the `SourceAddress`, `DestinationAddress`, and `DeviceAddress` event fields are selected, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields.

2. For each event field you need to read the IP address from, select the type of geographic data and the event field to which the geographic data should be written.

   You can use the **Add geodata attribute** button to add field pairs for **Geodata attribute** – **Event field to write to**. You can also configure different types of geographic data for one IP address to be written to different event fields. To delete a field pair, click ×.

   - In the **Geodata attribute** field, select which geographic data corresponding to the read IP address should be written to the event. Available geographic data attributes: **Country**, **Region**, **City**, **Longitude**, **Latitude**.

   - In the **Event field to write to**, select the event field which the selected geographic data attribute must be written to.

You can write identical geographic data attributes to different event fields. If you configure multiple geographic data attributes to be written to the same event field, the event will be enriched with the last mapping in the sequence.

4. Use the **Debug** drop-down list to indicate whether or not to enable logging of service operations. Logging is disabled by default.

5. In the **Filter** section, you can specify conditions to identify events that will be processed by the enrichment rule resource. You can select an existing filter from the drop-down list or **create** a new filter.

   Creating a filter in resources

   To create a filter:

   1. In the **Filter** drop-down list, select **Create new**.

   2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

   3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

   4. In the **Conditions** settings block, specify the conditions that the events must meet:

      a. Click the **Add condition** button.

      b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

      c. In the **operator** drop-down list, select an operator.

         Filter operators

         - **=**—the left operand equals the right operand.
         - **<**—the left operand is less than the right operand.
         - **<=**—the left operand is less than or equal to the right operand.
         - **>**—the left operand is greater than the right operand.
         - **>=**—the left operand is greater than or equal to the right operand.
         - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
         - **contains**—the left operand contains values of the right operand.
         - **startsWith**—the left operand starts with one of the values of the right operand.
         - **endsWith**—the left operand ends with one of the values of the right operand.
         - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

The new enrichment rule was added to the set of resources for the collector.

LDAP enrichment

To enable enrichment using LDAP:

1. Click **Add enrichment with LDAP data**.

   This opens the settings block for LDAP enrichment.

2. In the **LDAP accounts mapping** settings block, click the **New domain** button to specify the domain of the user accounts. You can specify multiple domains.

3. In the **LDAP mapping** table, define the rules for mapping KUMA fields to LDAP attributes:

   - In the **KUMA field** column, indicate the KUMA event field which data should be compared to LDAP attribute.

   - In the **LDAP attribute** column, specify the attribute that must be compared with the KUMA event field. The drop-down list contains standard attributes and can be augmented with **custom attributes**.

     Before configuring event enrichment using custom attributes, make sure that custom attributes are configured in AD.

     To enrich events with accounts using custom attributes:

     1. Add **Custom AD Account Attributes** in the LDAP connection settings.

        Standard **imported attributes from AD** cannot be added as custom attributes. For example, if you add the standard `accountExpires` attribute as a custom attribute, KUMA returns an error when saving the connection settings.

        The following account attributes can be requested from Active Directory:

        - `accountExpires`
        - `badPasswordTime`
        - `cn`
        - `company`
        - `department`
        - `displayName`
        - `distinguishedName`
        - `division`
        - `employeeID`
        - `ipaUniqueID`
        - `l`
        - `Mail`
        - `mailNickname`

- managedObjects
- manager
- memberOf (this attribute can be used for search during correlation)
- mobile
- objectGUID (this attribute always requested from Active Directory even if a user doesn't specify it)
- objectSID
- physicalDeliveryOfficeName
- sAMAccountName
- sAMAccountType
- sn
- streetAddress
- telephoneNumber
- title
- userAccountControl
- UserPrincipalName
- whenChanged
- whenCreated

After you add custom attributes in the LDAP connection settings, the **LDAP attribute** to receive drop-down list in the collector automatically includes the new attributes. Custom attributes are identified by a question mark next to the attribute name. If you added the same attribute for multiple domains, the attribute is listed only once in the drop-down list. You can view the domains by moving your cursor over the question mark. Domain names are displayed as links. If you click a link, the domain is automatically added to **LDAP accounts mapping** if it was not previously added.

If you deleted a custom attribute in the LDAP connection settings, manually delete the row containing the attribute from the mapping table in the collector. Account attribute information in KUMA is updated each time you import accounts.

2. Import accounts.
3. In the collector, in the **LDAP mapping** table, define the rules for mapping KUMA fields to LDAP attributes.
4. Restart the collector.

   After the collector is restarted, KUMA begins enriching events with accounts.

- In the **KUMA event field to write to** column, specify in which field of the KUMA event the ID of the user account imported from LDAP should be placed if the mapping was successful.

You can click the **Add row** button to add a string to the table, and can click the × button to remove a string. You can click the **Apply default mapping** button to fill the mapping table with standard values.

Event enrichment rules for data received from LDAP were added to the group of resources for the collector.

> If you add an enrichment to an existing collector using LDAP or change the enrichment settings, you must stop and restart the service.

Proceed to the next step of the Installation Wizard.

Step 7. Routing

This is an optional step of the Installation Wizard. On the **Routing** tab of the Installation Wizard, you can select or create destinations with settings indicating the forwarding destination of events processed by the collector. Typically, events from the collector are routed to two points: to the correlator to analyze and search for threats; and to the storage, both for storage and so that processed events can be viewed later. Events can be sent to other locations as needed. There can be more than one destination point.

To add an existing destination to a collector resource set:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. In the **Destination** drop-down list, select the necessary destination.

   The window name changes to **Edit destination**, and it displays the settings of the selected resource. To open the settings of a destination for editing in a new browser tab, click ↗.

3. Click **Save**.

The selected destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

To add a new destination resource to a collector resource set:

1. In the **Add destination** drop-down list, select the type of destination resource you want to add:

   - Select **Storage** if you want to configure forwarding of processed events to the storage.
   - Select **Correlator** if you want to configure forwarding of processed events to a correlator.
   - Select **Other** if you want to send events to other locations.

     > This type of resource includes correlator and storage services that were created in previous versions of the program.

   The **Add destination** window opens where you can specify parameters for events forwarding.

2. Specify the settings on the **Basic settings** tab:

   - In the **Destination** drop-down list, select **Create new**.
   - In the **Name** field, enter a unique name for the destination resource. The name must contain 1 to 128 Unicode characters.
   - Click the **Disabled** toggle button to specify whether events will be sent to this destination. By default, sending events is enabled.
   - Select the **Type** for the destination resource:
     - Select **storage** if you want to configure forwarding of processed events to the storage.
     - Select **correlator** if you want to configure forwarding of processed events to a correlator.
     - Select **nats-jetstream**, **tcp**, **http**, **kafka**, or **file** if you want to configure sending events to other locations.
   - Specify the **URL** to which events should be sent in the hostname:<API port> format.

     You can specify multiple destination addresses by clicking the **URL** button for all types except **nats-jetstream**, **file**, and **diode**.
   - For the **nats-jetstream** and **kafka** types, use the **Topic** field to specify which topic the data should be written to. The topic must contain Unicode characters. The Kafka topic is limited to 255 characters.

3. If necessary, specify the settings on the **Advanced settings** tab. The available settings vary based on the selected destination resource type:

   - **Compression** is a drop-down list where you can enable Snappy compression. By default, compression is **disabled**.
   - **Proxy** is a drop-down list for proxy server selection.
   - The **Buffer size** field is used to set buffer size (in bytes) for the destination. The default value is 1 MB, and the maximum value is 64 MB.
   - **Timeout** field is used to set the timeout (in seconds) for another service or component response. The default value is `30`.

- **Disk buffer size limit** field is used to specify the size of the disk buffer in bytes. The default size is 10 GB.
- **Cluster ID** is the ID of the NATS cluster.
- **TLS mode** is a drop-down list where you can specify the conditions for using TLS encryption:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—encryption is enabled, but without verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

  When using TLS, it is impossible to specify an IP address as a URL.
- **URL selection policy** is a drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified:
  - **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination.
  - **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again.
  - **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
- **Delimiter** is used to specify the character delimiting the events. By default, `\n` is used.
- **Path**—the file path if the **file** destination type is selected.
- **Buffer flush interval**—this field is used to set the time interval (in seconds) at which the data is sent to the destination. The default value is `100`.
- **Workers**—this field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- You can set health checks using the **Health check path** and **Health check timeout** fields. You can also disable health checks by selecting the **Health Check Disabled** check box.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
- The **Disk buffer disabled** drop-down list is used to enable or disable the use of a disk buffer. By default, the disk buffer is disabled.

  The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting.

  If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

- In the **Filter** section, you can specify the conditions to define events that will be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

  To create a filter:

  1. In the **Filter** drop-down list, select **Create new**.
  2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.
  3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.
  4. In the **Conditions** settings block, specify the conditions that the events must meet:

     a. Click the **Add condition** button.

     b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

     c. In the **operator** drop-down list, select an operator.

     Filter operators

     - **=**—the left operand equals the right operand.
     - **<**—the left operand is less than the right operand.
     - **<=**—the left operand is less than or equal to the right operand.
     - **>**—the left operand is greater than the right operand.
     - **>=**—the left operand is greater than or equal to the right operand.
     - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
     - **contains**—the left operand contains values of the right operand.
     - **startsWith**—the left operand starts with one of the values of the right operand.
     - **endsWith**—the left operand ends with one of the values of the right operand.
     - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
     - **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

       The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

4. Click **Save**.

The created destination is displayed on the Installation Wizard tab. A destination resource can be removed from the resource set by selecting it and clicking **Delete** in the opened window.

Proceed to the next step of the Installation Wizard.

Step 8. Setup validation

This is the required, final step of the Installation Wizard. At this step, KUMA creates a service resource set, and the Services are created automatically based on this set:

- The set of resources for the collector is displayed under **Resources** → **Collectors**. It can be used to create new collector services. When this set of resources changes, all services that operate based on this set of resources will start using the new parameters after the services restart. To do so, you can click the **Save and restart services** and **Save and update service configurations** buttons.

  A set of resources can be modified, copied, moved from one folder to another, deleted, imported, and exported, like other resources.

- Services are displayed in **Resources** → **Active services**. The services created using the Installation Wizard perform functions inside the KUMA program. To communicate with external parts of the network infrastructure, you need to install similar external services on the servers and assets intended for them. For example, an external collector service should be installed on a server intended as an events recipient, external storage services should be installed on servers that have a deployed ClickHouse service, and external agent services should be installed on the Windows assets that must both receive and forward Windows events.

To finish the Installation Wizard:

1. Click **Create and save service**.

   The **Setup validation** tab of the Installation Wizard displays a table of services created based on the set of resources selected in the Installation Wizard. The lower part of the window shows examples of commands that you must use to install external equivalents of these services on their intended servers and assets.

   For example:

   /opt/kaspersky/kuma/kuma collector --core https://kuma-example:<port used for communication with the KUMA Core> --id <service ID> --api.port <port used for communication with the service> --install

   > The "kuma" file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

   The port for communication with the KUMA Core, the service ID, and the port for communication with the service are added to the command automatically. You should also ensure the network connectivity of the KUMA system and open the ports used by its components if necessary.

2. Close the Wizard by clicking **Save collector**.

The collector service is created in KUMA. Now you will install a similar service on the server intended for receiving events.

> If a wmi or wec connector was selected for collectors, you must also install the automatically created KUMA agents.

Installing a collector in a KUMA network infrastructure

A collector consists of two parts: one part is created inside the KUMA Console, and the other part is installed on the network infrastructure server intended for receiving events. The second part of the collector is installed in the network infrastructure.

To install a collector:

1. Log in to the server where you want to install the service.

2. Create the /opt/kaspersky/kuma/ folder.

3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   Make sure the kuma file has sufficient rights to run. If the file is not executable, make it executable:

   ```
   sudo chmod +x /opt/kaspersky/kuma/kuma
   ```

4. Place the LICENSE file from the /kuma-ansible-installer/roles/kuma/files/ directory in the /opt/kaspersky/kuma/ directory and accept the license by running the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma license
   ```

5. Create the 'kuma' user:

   ```
   sudo useradd --system kuma && usermod -s /usr/bin/false kuma
   ```

6. Make the 'kuma' user the owner of the /opt/kaspersky/kuma directory and all files inside the directory:

   ```
   sudo chown -R kuma:kuma /opt/kaspersky/kuma/
   ```

7. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the KUMA Core server>:<port used by KUMA Core for internal communication (port 7210 is used by default)> --id <service ID copied from the KUMA Console> --api.port <port used for communication with the installed component>
   ```

   Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://test.kuma.com:7210 --id XXXX --api.port YYYY`

   If errors are detected as a result of the command execution, make sure that the settings are correct. For example, the availability of the required access level, network availability between the collector service and the Core, and the uniqueness of the selected API port. After fixing errors, continue installing the collector.

   If no errors were found, and the collector status in the KUMA Console is changed to *green*, stop the command execution and proceed to the next step.

   > The command can be copied at the last step of the installer wizard. It automatically specifies the address and port of the KUMA Core server, the identifier of the collector to be installed, and the port that the collector uses for communication.

   > When deploying several KUMA services on the same host, during the installation process you must specify unique ports for each component using the `--api.port <port>` parameter. The following setting values are used by default: `--api.port 7221`.

   > Before installation, ensure the network connectivity of KUMA components.

8. Run the command again by adding the `--install` key:

```
sudo /opt/kaspersky/kuma/kuma collector --core https://<FQDN of the KUMA Core server>:
<port used by KUMA Core for internal communication (port 7210 is used by default)> --
id <service ID copied from the KUMA Console> --api.port <port used for communication
with the installed component> --install
```

Example: `sudo /opt/kaspersky/kuma/kuma collector --core https://kuma.example.com:7210 --id XXXX --api.port YYYY --install`

9. Add KUMA collector port to firewall exclusions.

> For the program to run correctly, ensure that the KUMA components are able to interact with other components and programs over the network via the protocols and ports specified during the installation of the KUMA components.

The collector is installed. You can use it to receive data from an event source and forward it for processing.

Validating collector installation

To verify that the collector is ready to receive events:

1. In the KUMA Console, go to the **Resources** → **Active services** section.

2. Make sure that the collector you installed has the green status.

   If the status of the collector is not green, view the log of this service on the machine where it is installed, in the /opt/kaspersky/kuma/collector/<collector ID>/log/collector directory. Errors are logged regardless of whether debug mode is enabled or disabled.

If the collector is installed correctly and you are sure that data is coming from the event source, the table should display events when you search for events associated with the collector.

*To check for normalization errors using the **Events** section of the KUMA Console:*

1. Make sure that the Collector service is running.

2. Make sure that the event source is providing events to the KUMA.

3. Make sure that you selected **Only errors** in the **Keep raw event** drop-down list of the **Normalizer** resource in the **Resources** section of the KUMA Console.

4. In the **Events** section of KUMA, search for events with the following parameters:

   - `ServiceID = <ID of the collector to be checked>`

   - `Raw != ""`

   If any events are found with this search, it means that there are normalization errors and they should be investigated.

To check for normalization errors using the Grafana™ Dashboard:

1. Make sure that the Collector service is running.
2. Make sure that the event source is providing events to the KUMA.
3. Open the Metrics section and follow the KUMA Collectors link.
4. See if the Errors section of the Normalization widget displays any errors.

   If there are any errors, it means that there are normalization errors and they should be investigated.

For WEC and WMI collectors, you must ensure that unique ports are used to connect to their agents. This port is specified in the **Transport** section of Collector Installation Wizard.

Ensuring uninterrupted collector operation

An uninterrupted event stream from the event source to KUMA is important for protecting the network infrastructure. Continuity can be ensured though automatic forwarding of the event stream to a larger number of collectors:

- On the KUMA side, two or more identical collectors must be installed.
- On the event source side, you must configure control of event streams between collectors using third-party server load management tools, such as rsyslog or nginx.

With this configuration of the collectors in place, no incoming events will be lost if the collector server is unavailable for any reason.

Please keep in mind that when the event stream switches between collectors, each collector will aggregate events separately.

If the KUMA collector fails to start, and its log includes the "panic: runtime error: slice bounds out of range [8:0]" error:

1. Stop the collector.

   ```
   sudo systemctl stop kuma-collector-<collector ID>
   ```

2. Delete the DNS enrichment cache files.

   ```
   sudo rm -rf /opt/kaspersky/kuma/collector/<collector ID>/cache/enrichment/DNS-*
   ```

3. Delete the event cache files (disk buffer). Run the command only if you can afford to jettison the events in the disk buffers of the collector.

   ```
   sudo rm -rf /opt/kaspersky/kuma/collector/<collector ID>/buffers/*
   ```

4. Start the collector service.

   ```
   sudo systemctl start kuma-collector-<collector ID>
   ```

Event stream control using rsyslog

To enable rsyslog event stream control on the event source server:

1. Create two or more identical collectors that you want to use to ensure uninterrupted reception of events.

2. Install rsyslog on the event source server (see the rsyslog documentation).

3. Add rules for forwarding the event stream between collectors to the configuration file /etc/rsyslog.conf:

```
*. * @@ <main collector server FQDN>: <port for incoming events>
$ActionExecOnlyWhenPreviousIsSuspended on
*. * @@ <backup collector server FQDN>: <port for incoming events>
$ActionExecOnlyWhenPreviousIsSuspended off
```

Example configuration file

Example configuration file specifying one primary and two backup collectors. The collectors are configured to receive events on TCP port 5140.

```
*.* @@kuma-collector-01.example.com:5140
$ActionExecOnlyWhenPreviousIsSuspended on
& @@kuma-collector-02.example.com:5140
& @@kuma-collector-03.example.com:5140
$ActionExecOnlyWhenPreviousIsSuspended off
```

4. Restart rsyslog by running the following command:

   `systemctl restart rsyslog`.

Event stream control is now enabled on the event source server.

Event stream control using nginx

To control event stream using nginx, you need to create and configure a ngnix server to receive events from the event source and then forward these to collectors.

To enable nginx event stream control on the event source server:

1. Create two or more identical collectors that you want to use to ensure uninterrupted reception of events.

2. Install nginx on the server intended for event stream control.

   - Installation command in Oracle Linux 8.6:

     `$sudo dnf install nginx`

   - Installation command in Ubuntu 20.4:

     `$sudo apt-get install nginx`

When installing from sources, you must compile with the parameter `-with-stream` option:
`$ sudo ./configure -with-stream -without-http_rewrite_module -without-http_gzip_module`

3. On the nginx server, add the stream module to the nginx.conf configuration file that contains the rules for forwarding the stream of events between collectors.

   Example stream module

   Example module in which event stream is distributed between the collectors kuma-collector-01.example.com and kuma-collector-02.example.com, which receive events via TCP on port 5140 and via UDP on port 5141. Balancing uses the nginx.example.com ngnix server.

```
stream {
 upstream syslog_tcp {
server kuma-collector-1.example.com:5140;
server kuma-collector-2.example.com:5140;
}
upstream syslog_udp {
server kuma-collector-1.example.com:5141;
server kuma-collector-2.example.com:5141;
}
 server {
listen nginx.example.com:5140;
proxy_pass syslog_tcp;
}
server {
listen nginx.example.com:5141 udp;
proxy_pass syslog_udp;
proxy_responses 0;
}
}
 worker_rlimit_nofile 1000000;
events {
worker_connections 20000;
}
# worker_rlimit_nofile is the limit on the number of open files (RLIMIT_NOFILE) for workers. This is used to raise the limit without restarting the main process.
# worker_connections is the maximum number of connections that a worker can open simultaneously.
```

4. Restart nginx by running the following command:

   `systemctl restart nginx`

5. On the event source server, forward events to the ngnix server.

   Event stream control is now enabled on the event source server.

Nginx Plus may be required to fine-tune balancing, but certain balancing methods, such as Round Robin and Least Connections, are available in the base version of ngnix.

For more details on configuring nginx, please refer to the nginx documentation.

Predefined collectors

The predefined collectors listed in the table below are included in the OSMP distribution kit.

Predefined collectors

Name	Description
[OOTB] CEF	Collects CEF events received over the TCP protocol.
[OOTB] KSC	Collects events from Kaspersky Security Center over the Syslog TCP protocol.
[OOTB] KSC SQL	Collects events from Kaspersky Security Center using an MS SQL database query.
[OOTB] Syslog	Collects events via the Syslog protocol.
[OOTB] Syslog-CEF	Collects CEF events that arrive over the UDP protocol and have a Syslog header.

Creating an agent

A KUMA agent consists of two parts: one part is created inside the KUMA Console, and the second part is installed on a server or on an asset in the network infrastructure.

An agent is created in several steps:

1. **Creating a set of resources for an agent in the KUMA Console**
2. **Creating an agent service in the KUMA Console**
3. **Installing the server portion of the agent to the asset that will forward messages**

A KUMA agent for Windows assets can be created automatically when you create a collector with the wmi or wec transport type. Although the set of resources and service of these agents are created in the Collector Installation Wizard, they must still be installed to the asset that will be used to forward a message.

Creating a set of resources for an agent

In the KUMA Console, an agent service is created based on the set of resources for an agent that unites connectors and destinations.

To create a set of resources for an agent in the KUMA Console:

1. In the KUMA Console, under **Resources** → **Agents**, click **Add agent**.

   This opens a window for creating an agent with the **Base settings** tab active.

2. Specify the settings on the **Base settings** tab:

   - In the **Agent name** field, enter a unique name for the created service. The name must contain 1 to 128 Unicode characters.
   - In the **Tenant** drop-down list, select the tenant that will own the storage.

- If necessary, move the **Debug** toggle switch to the active position to enable logging of service operations.
- You can optionally add up to 256 Unicode characters describing the service in the **Description** field.

3. Click the plus icon (+) to create a connection for the agent and switch to the added **Connection <number>** tab.

   You can remove tabs by clicking the plus icon (×).

4. In the **Connector** group of settings, add a connector:

   - If you want to select an existing connector, select it from the drop-down list.
   - If you want to create a new connector, select **Create new** in the drop-down list and specify the following settings:
     - Specify the connector name in the **Name** field. The name must contain 1 to 128 Unicode characters.
     - In the **Type** drop-down list, select the connector type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of connector:
       - tcp
       - udp
       - nats-jetstream
       - kafka
       - http
       - file
       - ftp
       - nfs
       - wmi
       - wec
       - snmp

       The agent type is determined by the connector that is used in the agent. The only exception is for agents with a destination of the diode type. These agents are considered to be diode agents.

       When using the **tcp** or **udp** connector type at the normalization stage, IP addresses of the assets from which the events were received will be written in the DeviceAddress event field if it is empty.

> The ability to edit previously created wec or wmi connections in agents, collectors, and connectors is limited. You can change the connection type from **wec** to **wmi** and vice versa, but you cannot change the **wec** or **wmi** connection to any other connection type. At the same time, when editing other connection types, you cannot select the **wec** or **wmi** types. You can create connections without any restrictions on the types of connectors.

- You can optionally add up to 4,000 Unicode characters describing the resource in the **Description** field.

The connector is added to the selected connection of the agent's set of resources. The created connector is only available in this resource set and is not displayed in the web interface **Resources** → **Connectors section**.

5. In the **Destinations** group of settings, add a destination.

   - If you want to select an existing destination, select it from the drop-down list.
   - If you want to create a new destination, select **Create new** in the drop-down list and specify the following settings:
     - Specify the destination name in the **Name** field. The name must contain 1 to 128 Unicode characters.
     - In the **Type** drop-down list, select the destination type and specify its settings on the **Basic settings** and **Advanced settings** tabs. The available settings depend on the selected type of destination:
       - **nats-jetstream**—used for NATS communications.
       - **tcp**—used for communications over TCP.
       - **http**—used for HTTP communications.
       - **diode**—used to transmit events using a data diode.
       - **kafka**—used for Kafka communications.
       - **file**—used for writing to a file.
   - You can optionally add up to 4,000 Unicode characters describing the resource in the **Description** field.

   > The advanced settings for an agent destination (such as TLS mode and compression) must match the advanced destination settings for the collector that you want to link to the agent.

   There can be more than one destination point. You can add them by clicking the **Add destination** button and can remove them by clicking the ✕ button.

6. Repeat steps 3–5 for each agent connection that you want to create.

7. Click **Save**.

The set of resources for the agent is created and displayed under **Resources** → **Agents**. Now you can create an agent service in KUMA.

Creating an agent service in the KUMA Console

When a set of resources is created for an agent, you can proceed to create an agent service in KUMA.

To create an agent service in the KUMA Console:

1. In the KUMA Console, under **Resources** → **Active services**, click **Add service**.

2. In the opened **Choose a service** window, select the set of resources that was just created for the agent and click **Create service**.

   The agent service is created in the KUMA Console and is displayed under **Resources** → **Active services**. Now agent services must be installed to each asset from which you want to forward data to the collector. A service ID is used during installation.

Installing an agent in a KUMA network infrastructure

When an agent service is created in KUMA, you can proceed to installation of the agent to the network infrastructure assets that will be used to forward data to a collector.

> Prior to installation, verify the network connectivity of the system and open the ports used by its components.

Installing a KUMA agent on Linux assets

KUMA agent installed on Linux devices stops when you close the terminal or restart the server. To avoid starting the agents manually, we recommend installing the agent by using a system that automatically starts applications when the server is restarted, such as Supervisor. To start the agents automatically, define the automatic startup and automatic restart settings in the configuration file. For more details on configuring settings, please refer to the official documentation of automatic application startup systems. An example of configuring settings in Supervisor, which you can adapt to your needs:

```
[program:agent_<agent name>] command=sudo /opt/kaspersky/kuma/kuma agent --core
https://<KUMA Core server FQDN>:<port used by KUMA Core>

autostart=true

autorestart=true
```

To install a KUMA agent to a Linux asset:

1. Log in to the server where you want to install the service.

2. Create the following directories:

   - /opt/kaspersky/kuma/

   - /opt/kaspersky/agent/

3. Copy the "kuma" file to the /opt/kaspersky/kuma/ folder. The file is located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

   Make sure the kuma file has sufficient rights to run.

4. Execute the following command:

   ```
   sudo /opt/kaspersky/kuma/kuma agent --core https://<KUMA Core server FQDN>:<port used by KUMA Core server for internal communication (port 7210 by default)> --id <service ID copied from the KUMA console> --wd <path to the directory that will contain the files of the installed agent. If this flag is not specified, the files will be stored in the directory where the kuma file is located>
   ```

   Example: `sudo /opt/kaspersky/kuma/kuma agent --core https://kuma.example.com:7210 --id XXXX --wd /opt/kaspersky/kuma/agent/XXXX`

The KUMA agent is installed on the Linux asset. The agent forwards data to KUMA, and you can set up a collector to receive this data.

Installing an Open Single Management Platform agent on Windows assets

Prior to installing an Open Single Management Platform agent to a Windows asset, the server administrator must create a user account with the EventLogReaders and Log on as a service permissions on the Windows asset. This user account must be used to start the agent.
If you want to run the agent under a local account, you will need administrator rights and Log on as a service. If you want to perform the collection remotely and only read logs under a domain account, EventLogReaders rights are sufficient.

To install an Open Single Management Platform agent to a Windows asset:

1. Copy the kuma.exe file to a folder on the Windows asset. `C:\Users\<User name>\Desktop\Open Single Management Platform` folder is recommended for installation.

   The kuma.exe file is located inside the installer in the /kuma-ansible-installer/roles/kuma/files/ folder.

2. Start the Command Prompt on the Windows asset with Administrator privileges and locate the folder containing the kuma.exe file.

3. Execute the following command:

   ```
   kuma agent --core https://<fully qualified domain name of the Open Single Management Platform Core server>:<port used by the Open Single Management Platform Core server for internal communications (port 7210 by default)> --id <ID of the agent service that was created in Open Single Management Platform> --user <name of the user account used to run the agent, including the domain> --install [--accept-eula]
   ```

   To run the agent, you need to accept the End User License Agreement. You can add the `--accept-eula` option to the command to automatically accept the End User License Agreement during agent installation. This lets you perform the installation non-interactively. If you do not specify this option, you will need to accept or reject the License Agreement manually during the installation of the Open Single Management Platform agent.

Examples:

- Installing the Open Single Management Platform agent without automatically accepting the End User License Agreement:

  ```
  kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install
  ```

- Installing the Open Single Management Platform agent with automatic acceptance of the End User License Agreement:

  ```
  kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install --accept-eula
  ```

By using the `--accept-eula` option during the installation of the KUMA agent, you confirm that you agree with and accept the terms and conditions of the End User License Agreement.

You can get help information by executing the `kuma help agent` command.

4. If you started the installation of the agent without automatically accepting the End User License Agreement, during the installation process, you will be prompted to read the text of the End User License Agreement and you will have the opportunity to accept or reject the agreement.

5. If you chose installation with the automatic acceptance of the End User License Agreement and want to read the text of the End User License Agreement, or if the text of the End User License Agreement was not automatically provided to you during the installation process, run the following command:

   ```
   kuma.exe license --show
   ```

   If you want to accept the End User License Agreement, run the following command and press `y`:

   ```
   kuma.exe license
   ```

6. Specify the password for the agent's user in one the following ways:

   - Manually.

   - From a file when installing the agent.

     To avoid entering the password manually, you can add the password to a .TXT file and place the file in the same directory as the executable file of the agent.

     Specify the file containing the password in the command. For example:

     ```
     kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install < <name of the file with the password>.txt
     ```

     All data from the password file (for example, password.txt) is redirected to the command as the password, excluding end-of-line characters (\n\r, \n).

     Path for storing the agent folder after installation: C:\ProgramData\Kaspersky Lab\Open Single Management Platform\agent\${agent_id}.

If you did not accept the End User License Agreement (EULA) before installing the agent with the `<<password file name>`.txt flag, an error message is displayed saying that the agreement was not accepted.

You can accept the EULA in the following ways:

a. In the password file, you can specify the EULA acceptance flag. To achieve this, the file with the password must contain two lines: the first line containing "y", which serves to assert EULA acceptance, and the second line containing the password. For example:

```
y
<password>
```

The agent installation command looks like this:

```
kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --install < <name of the file with the password>.txt
```

In this case, you will not need to additionally accept the EULA when installing the agent.

If the EULA has been accepted before running the command with the password file, the first line in the file is interpreted as the password and the installation is aborted with the `Access denied` error.

b. You can accept the EULA and perform a silent installation at the same time by using the --accept-eula flag.

The agent installation command looks like this:

```
kuma agent --core https://kuma.example.com:7210 --id XXXXX --user domain\username --accept-eula --install < <name of the file with the password>.txt
```

Make sure that only the password is specified in the file, on the first line, otherwise an error will occur.

The `C:\ProgramData\Kaspersky Lab\Open Single Management Platform\agent\<agent ID>` folder is created and the Open Single Management Platform agent service is installed in it. The agent forwards Windows events to Open Single Management Platform, and you can set up a collector to receive them.

When the agent service is installed, it starts automatically. The service is also configured to restart in case of any failures. The agent can be restarted from the Open Single Management Platform web interface, but only when the service is active. Otherwise, the service needs to be manually restarted on the Windows asset.

Removing an Open Single Management Platform agent from Windows assets

To remove a KUMA agent from a Windows asset:

1. Start the Command Prompt on the Windows machine with Administrator privileges and locate the folder with kuma.exe file.

2. Run any of the commands below:

   - `kuma.exe agent --cfg <path to agent configuration file> --uninstall`

   - `kuma.exe agent --id <ID of agent service that was created in KUMA> --uninstall`

   The specified KUMA agent is removed from the Windows asset. Windows events are no longer sent to KUMA.

When configuring services, you can check the configuration for errors before installation by running the agent with the following command:

```
kuma agent --core https://<fully qualified domain name of the Open Single Management
Platform Core server>:<port used by the Open Single Management Platform Core server for
internal communications (port 7210 by default)> --id <ID of the agent service that was
created in Open Single Management Platform> --user <name of the user account used to run
the agent, including the domain>
```

Automatically created agents

When creating a collector with wec or wmi connectors, agents are automatically created for receiving Windows events.

Automatically created agents have the following special conditions:

- Automatically created agents can have only one connection.
- Automatically created agents are displayed under **Resources** → **Agents**, and `auto created` is indicated at the end of their name. Agents can be reviewed or deleted.
- The settings of automatically created agents are defined automatically based on the collector settings from the **Connect event sources** and **Transport** sections. You can change the settings only for a collector that has a created agent.
- The description of an automatically created agent is taken from the collector description in the **Connect event sources** section.
- Debugging of an automatically created agent is enabled and disabled in the **Connect event sources** section of the collector.
- When deleting a collector with an automatically created agent, you will be prompted to choose whether to delete the collector together with the agent or to just delete the collector. When deleting only the collector, the agent will become available for editing.
- When deleting automatically created agents, the type of collector changes to **http**, and the connection address is deleted from the **URL** field of the collector.
- If at least one Windows log name in wec or wmi connector is specified incorrectly, the agent will not receive events from any Windows log listed in the connector. At the same time the agent status will be green. Attempts to receive events will be repeated every 60 seconds, and error messages will be added to the service log.

In the KUMA interface, automatically created agents appear at the same time when the collector is created. However, they must still be installed on the asset that will be used to forward a message.

Update agents

When updating KUMA versions, the WMI and WEC agents installed on remote machines must also be updated.

To update the agent, use an administrator account and follow these steps:

1. In the KUMA Console, in the **Resources** → **Active services** → **Agents** section, select the agent that you want to update and copy its ID.

   You need the ID to install the new agent with the same ID after removing the old agent.

2. In Windows, in the **Services** section, open the agent and click **Stop**.

3. On the command line, go to the folder where the agent is installed and run the command to remove the agent from the server.

   kuma.exe agent --id <ID of agent service that was created in KUMA> --uninstall

4. Place the new agent in the same folder.

5. On the command line, go to the folder with the new agent and from that folder, run the installation command using the agent ID from step 1.

   kuma agent --core https://<fullly qualified domain name of the KUMA Core server>:<port used by the KUMA Core server for internal communications (port 7210 by default)> --id <ID of the agent service that was created in KUMA> --user <name of the user account used to run the agent, including the domain> --install

The agent is updated.

Transferring events from isolated network segments to KUMA

Data transfer scenario

Data diodes can be used to transfer events from isolated network segments to KUMA. Data transfer is organized as follows:

1. KUMA agent that is Installed on a standalone server, with a **diode** destination receives events and moves them to a directory from which the data diode will pick up the events.

   The agent accumulates events in a buffer until it overflows or for a user-defined period after the last write to disk. The events are then written to a file in the temporary directory of the agent. The file is moved to the directory processed by the data diode; its name is a combination of the file contents hash (SHA256) and the file creation time.

2. The data diode moves files from the isolated server directory to the external server directory.

3. A KUMA collector with a **diode** connector installed on an external server reads and processes events from the files of the directory where the data diode places files.

   After all events are read from a file, it is automatically deleted. Before reading events, the contents of files are verified based on the hash in the file name. If the contents fail verification, the file is deleted.

In the described scenario, the KUMA components are responsible for moving events to a specific directory within the isolated segment and for receiving events from a specific directory in the external network segment. The data diode transfers files containing events from the directory of the isolated network segment to the directory of the external network segment.

For each data source within an isolated network segment, you must create its own KUMA collector and agent, and configure the data diode to work with separate directories.

Configuring KUMA components

Configuring KUMA components for transferring data from isolated network segments consists of the following steps:

1. Creating a collector service in the external network segment.

   At this step, you must create and install a collector to receive and process the files that the data diode will transfer from the isolated network segment. You can use the Collector Installation Wizard to create the collector and all the resources it requires.

   At the **Transport** step, you must select or create a connector of the **diode** type. In the connector, you must specify the directory to which the data diode will move files from the isolated network segment.

   The user "kuma" that runs the collector must have read/write/delete permissions in the directory to which the data diode moves data from the isolated network segment.

2. Creating a set of resources for a KUMA agent.

   At this step, you must create a set of resources for the KUMA agent that will receive events in an isolated network segment and prepare them for transferring to the data diode. The diode agent resource set has the following requirements:

   - The destination in the agent must have the **diode** type. In this resource, you must specify the directory from which the data diode will move files to the external network segment.
   - You cannot select connectors of the **sql** or **netflow** types for the diode agent.
   - TLS mode must be disabled in the connector of the diode agent.

3. Downloading the agent configuration file as JSON file.

   a. The set of agent resources from a diode-type destination must be downloaded as a JSON file.

   b. If secret resources were used in the agent resource set, you must manually add the secret data to the configuration file.

4. Installing the KUMA agent service in the isolated network segment.

   At this step, you must install the agent in an isolated network segment based on the agent configuration file that was created at the previous step. It can be installed to Linux and Windows devices.

Configuring a data diode

The data diode must be configured as follows:

- Data must be transferred atomically from the directory of the isolated server (where the KUMA agent places the data) to the directory of the external server (where the KUMA collector reads the data).
- The transferred files must be deleted from the isolated server.

For information on configuring the data diode, please refer to the documentation for the data diode used in your organization.

Special considerations

When working with isolated network segments, operations with SQL and NetFlow are not supported.

When using the scenario described above, the agent cannot be administered through the KUMA Console because it resides in an isolated network segment. Such agents are not displayed in the list of active KUMA services.

Diode agent configuration file

A created set of agent resources with a diode-type destination can be downloaded as a configuration file. This file is used when installing the agent in an isolated network segment.

To download the configuration file:

In the KUMA Console, under **Resources** → **Agents**, select the relevant set of agent resources with a 'diode' destination and click **Download config**.

The agent settings configuration is downloaded as a JSON file based on the settings of your browser. Secrets used in the agent resource set are downloaded empty. Their IDs are specified in the file in the `secrets` section. To use a configuration file to install an agent in an isolated network segment, you must manually add secrets to the configuration file (for example, specify the URL and passwords used in the agent connector to receive events).

You must use an access control list (ACL) to configure permissions to access the file on the server where the agent will be installed. File read access must be available to the user account that will run the diode agent.

Below is an example of a diode agent configuration file with a kafka connector.

```
{
  "config": {
    "id": "<ID of the set of agent resources>",
    "name": "<name of the set of agent resources>",
    "proxyConfigs": [
      {
        "connector": {
          "id": "<ID of the connector. This example shows a kafka-type connector, but other types of connectors can also be used in a diode agent.
If a connector is created directly in the set of agent resources, the ID is not defined.>",
          "name": "<name of the connector>",
          "kind": "kafka",
          "connections": [
            {
              "kind": "kafka",
              "urls": [
                "localhost:9093"
              ],
              "host": "",
              "port": "",
              "secretID": "<ID of the secret>",
              "clusterID": "",
              "tlsMode": "",
              "proxy": null,
              "rps": 0,
              "maxConns": 0,
              "urlPolicy": "",
              "version": "",
              "identityColumn": "",
              "identitySeed": "",
              "pollInterval": 0,
              "query": "",
              "stateID": "",
              "certificateSecretID": "",
              "authMode": "pfx",
              "secretTemplateKind": "",
              "certSecretTemplateKind": ""
            }
          ],
          "topic": "<kafka topic name>",
          "groupID": "<kafka group ID>",
          "delimiter": "",
          "bufferSize": 0,
          "characterEncoding": "",
          "query": "",
          "pollInterval": 0,
          "workers": 0,
          "compression": "",
          "debug": false,
          "logs": [],
          "defaultSecretID": "",
          "snmpParameters": [
            {
              "name": "-",
              "oid": "",
              "key": ""
            }
          ],
          "remoteLogs": null,
          "defaultSecretTemplateKind": ""
        },
        "destinations": [
          {
            "id": "<ID of the destination. If the destination is created directly in the set of agent resources, the ID is not defined.>",
            "name": "<destination name>",
            "kind": "diode",
            "connection": {
              "kind": "file",
              "urls": [
                "<path to the directory where the destination should place events that the data diode will transmit from the isolated network
segment>",
                "<path to the temporary directory in which events are placed to prepare for data transmission by the diode>"
              ],
              "host": "",
              "port": "",
              "secretID": "",
              "clusterID": "",
              "tlsMode": "",
              "proxy": null,
```

```
                "rps": 0,
                "maxConns": 0,
                "urlPolicy": "",
                "version": "",
                "identityColumn": "",
                "identitySeed": "",
                "pollInterval": 0,
                "query": "",
                "stateID": "",
                "certificateSecretID": "",
                "authMode": "",
                "secretTemplateKind": "",
                "certSecretTemplateKind": ""
              },
              "topic": "",
              "bufferSize": 0,
              "flushInterval": 0,
              "diskBufferDisabled": false,
              "diskBufferSizeLimit": 0,
              "healthCheckPath": "",
              "healthCheckTimeout": 0,
              "healthCheckDisabled": false,
              "timeout": 0,
              "workers": 0,
              "delimiter": "",
              "debug": false,
              "disabled": false,
              "compression": "",
              "filter": null,
              "path": ""
            }
          ]
        }
      ],
      "workers": 0,
      "debug": false
    },
    "secrets": {
      "<secret ID>": {
        "pfx": "<encrypted pfx key>",
        "pfxPassword": "<password to the encrypted pfx key. The changeit value is exported from KUMA instead of the actual password. In the configuration file, you must manually specify the contents of secrets>"
      }
    },
    "tenantID": "<ID of the tenant>"
}
```


Description of secret fields

Secret fields

Field name	Type	Description
`user`	string	User name
`password`	string	Password
`token`	string	Token
`urls`	array of strings	URL list
`publicKey`	string	Public key (used in PKI)
`privateKey`	string	Private key (used in PKI)
`pfx`	string containing the base64-encoded pfx file	Base64-encoded contents of the PFX file. In Linux, you can get the base64 encoding of a file by running the following command: `base64 -w0 src > dst`
`pfxPassword`	string	Password of the PFX
`securityLevel`	string	Used in snmp3. Possible values: `NoAuthNoPriv`, `AuthNoPriv`, `AuthPriv`

`community`	string	Used in snmp1
`authProtocol`	string	Used in snmp3. Possible values: `MD5`, `SHA`, `SHA224`, `SHA256`, `SHA384`, `SHA512`
`privacyProtocol`	string	Used in snmp3. Possible values: `DES`, `AES`
`privacyPassword`	string	Used in snmp3
`certificate`	string containing the base64-encoded pem file	Base64-encoded contents of the PEM file. In Linux, you can get the base64 encoding of a file by running the following command: `base64 -w0 src > dst`

Installing Linux Agent in an isolated network segment

To install a KUMA agent to a Linux device in an isolated network segment:

1. Place the following files on the Linux server in an isolated network segment that will be used by the agent to receive events and from which the data diode will move files to the external network segment:

   - Agent configuration file.

     You must use an access control list (ACL) to configure access permissions for the configuration file so that only the KUMA user will have file read access.

   - Executive file /opt/kaspersky/kuma/kuma (the "kuma" file can located in the installer in the /kuma-ansible-installer/roles/kuma/files/ folder).

2. Execute the following command:

   ```
   sudo ./kuma agent --cfg <path to the agent configuration file> --wd <path to the directory where the files of the agent being installed will reside. If this flag is not specified, the files will be stored in the directory where the kuma file is located>
   ```

   The agent service is installed and running on the server in an isolated network segment. It receives events and relays them to the data diode so that they can be sent to an external network segment.

Installing Windows Agent in an isolated network segment

Prior to installing a KUMA agent to a Windows asset, the server administrator must create a user account with the EventLogReaders and Log on as a service permissions on the Windows asset. This user account must be used to start the agent.

To install a KUMA agent to a Windows device in an isolated network segment:

1. Place the following files on the Window server in an isolated network segment that will be used by the agent to receive events and from which the data diode will move files to the external network segment:

   - Agent configuration file.

     > You must use an access control list (ACL) to configure access permissions for the configuration file so that the file can only be read by the user account that will run the agent.

   - Kuma.exe executable file. This file can be found inside the installer in the /kuma-ansible-installer/roles/kuma/files/ directory.

   It is recommended to use the `C:\Users\<user name>\Desktop\KUMA` folder.

2. Start the Command Prompt on the Windows asset with Administrator privileges and locate the folder containing the kuma.exe file.

3. Execute the following command:

   ```
   kuma.exe agent --cfg <path to the agent configuration file> --user <user name that will run the agent, including the domain> --install
   ```

   > You can get installer Help information by running the following command:

   ```
   kuma.exe help agent
   ```

4. Enter the password of the user account used to run the agent.

   The `C:\Program Files\Kaspersky Lab\KUMA\agent\<Agent ID>` folder is created in which the KUMA agent service is installed. The agent moves events to the folder so that they can be processed by the data diode.

   > When installing the agent, the agent configuration file is moved to the directory C:\Program Files\Kaspersky Lab\KUMA\agent\<agent ID specified in the configuration file>. The kuma.exe file is moved to the C:\Program Files\Kaspersky Lab\KUMA directory.

   > When installing an agent, its configuration file must not be located in the directory where the agent is installed.

When the agent service is installed, it starts automatically. The service is also configured to restart in case of any failures.

Removing a KUMA agent from Windows assets

To remove a KUMA agent from a Windows asset:

1. Start the Command Prompt on the Windows machine with Administrator privileges and locate the folder with kuma.exe file.

2. Run any of the commands below:

   - `kuma.exe agent --cfg <path to agent configuration file> --uninstall`

   - `kuma.exe agent --id <ID of agent service that was created in KUMA> --uninstall`

   The specified KUMA agent is removed from the Windows asset. Windows events are no longer sent to KUMA.

When configuring services, you can check the configuration for errors before installation by running the agent with the following command:

```
kuma.exe agent --cfg <path to agent configuration file>
```

Transferring events from Windows machines to KUMA

To transfer events from Windows machines to KUMA, a combination of a KUMA agent and a KUMA collector is used. Data transfer is organized as follows:

1. The KUMA agent installed on the machine receives Windows events:

   - Using the WEC connector: the agent receives events arriving at the host under a subscription, as well as the server logs.

   - Using the WMI connector: the agent connects to remote servers specified in the configuration and receives events.

2. The agent sends events (without preprocessing) to the KUMA collector specified in the destination.

   You can configure the agent so that different logs are sent to different collectors.

3. The collector receives events from the agent, performs a full event processing cycle, and sends the processed events to the destination.

Receiving events from the WEC agent is recommended when using centralized gathering of events from Windows hosts using Windows Event Forwarding (WEF). The agent must be installed on the server that collects events; it acts as the Windows Event Collector (WEC). We do not recommend installing KUMA agents on every endpoint host from which you want to receive events.

The process of configuring the receipt of events using the WEC Agent is described in detail in the appendix: Configuring receipt of events from Windows devices using KUMA Agent (WEC).

For details about the Windows Event Forwarding technology, please refer to the official Microsoft documentation.

We recommend receiving events using the WMI agent in the following cases:

- If it is not possible to use the WEF technology to implement centralized gathering of events, and at the same time, installation of third-party software (for example, the KUMA agent) on the event source server is prohibited.
- If you need to obtain events from a small number of hosts — no more than 500 hosts per one KUMA agent.

For connecting Windows logs as an event source, we recommend using the **Add event source** wizard. When using a wizard to create a collector with WEC or WMI connectors, agents are automatically created for receiving Windows events. You can also manually create the resources necessary for collecting Windows events.

An agent and a collector for receiving Windows events are created and installed in several stages:

1. **Creating a set of resources for an agent**

   Agent connector:

   When creating an agent, on the **Connection** tab, you must create or select a connector of the WEC or WMI type.

   If at least one Windows log name in a WEC or WMI connector is specified incorrectly, the agent will receive events from all Windows logs listed in the connector, except the problematic log. At the same time the agent status will be green. Attempts to receive events will be repeated every 60 seconds, and error messages will be added to the service log.

   Agent destination:

   The type of agent destination depends on the data transfer method you use: nats, tcp, http, diode, kafka, file.

   You must use the `\0` value as the destination separator.

   The advanced settings for the agent destination (such as separator, compression and TLS mode) must match the advanced destination settings for the collector connector that you want to link to the agent.

2. **Creating an agent service in the KUMA Console**

3. **Installing the KUMA agent on the Windows machine from which you want to receive Windows events.**

   Before installation, make sure that the system components have access to the network and open the necessary network ports:

   - Port 7210, TCP: from server with collectors to the Core.
   - Port 7210, TCP: from agent server to the Core.
   - The port configured in the **URL** field when the connector was created: from the agent server to the server with the collector.

4. **Creating and installing KUMA collector.**

   When creating a set of collectors, at the **Transport** step, you must create or select a connector that the collector will use to receive events from the agent. Connector type must match the type of the agent destination.

   The advanced settings of the connector (such as delimiter, compression, and TLS mode) must match the advanced settings of the agent destination that you want to link to the agent.

For some playbooks to work correctly, you may need to configure additional enrichment of the collector.

To edit enrichment rule settings in the KUMA collector:

1. Add an enrichment rule by clicking Add enrichment rule and specify the following information in the corresponding fields:

   - **Name**: Specify an arbitrary name for the rule.
   - **Source kind**: dns.
   - **URL**: IP address of the domain controller.
   - **Requests per second**: 5.
   - **Workers**: 2.
   - **Cache TTL**: 3600.

2. Add an enrichment rule by clicking Add enrichment rule and do the following:

   a. Fill in the following fields:

      - **Name**: Specify an arbitrary name for the rule.
      - **Source kind**: event.
      - **Source field**: DestinationNTDomain.
      - **Target field**: DestinationNTDomain.

   b. Click **Add conversion** and specify the following information in the corresponding fields:

      - **Type**: append.
      - **Constant**: `.RU`.
      - **Type**: replace.
      - **Chars**: `RU.RU`.
      - **With chars**: `RU`.

3. Repeat the substeps from step 2 and specify SourceNTDomain as the **Source field** and **Target field**.

4. Add enrichment with LDAP data and do the following:

   - Under **LDAP accounts mapping**, specify the name of the domain controller.
   - Click **Apply default mapping** to fill the mapping table with standard values.

AI services

This section provides information about the AI services that KUMA works with.

Analyzing alerts and incidents with KIRA

You can use Kaspersky Investigation & Response Assistant (KIRA) to analyze alerts and incidents. As a result of the analysis, you receive an automatically generated description of the alert or incident, including a short summary, a list of observables, an event timeline, an alert or incident report, and mitigation recommendations.

This functionality accelerates the investigation and initial analysis of alerts or security incidents and is available regardless of the event types included in the alert or incident.

The functionality is available in the RU region if the following conditions are satisfied:

- There is a current license covering the AI module.

  If the license has expired, the analysis results remain available through tasks during the lifetime of the cache, that is, for 14 days from the moment the result is cached.

- A certificate was uploaded when configuring the KIRA integration.

- The user has **Write** rights for the **Alerts and Incidents** functional area.

Configuring integration with KIRA

For integration with KIRA, you will need a certificate. You can get the certificate file in PFX format, packed in the <customer name>.ZIP archive, and the password for the certificate from Technical Support.

To configure integration with KIRA:

1. Get a license with the AI module and activate the functionality in KUMA.

2. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

3. Under **Integrations**, click **AI services**.

4. In the open window, go to the **Kaspersky Investigation & Response Assistant** tab.

5. Enable integration with KIRA by enabling the **Status** toggle switch.

   The toggle switch is turned off by default.

6. Click **Select file** and download the <customer name> .ZIP archive, which contains the certificate in PFX format.

7. In the **Certificate password** field, enter the password.

8. In the **Proxy** drop-down list, select a previously created resource or create a new resource.

9. In the drop-down list, select the KIRA response language, Russian or English. KUMA sends and stores KIRA responses in the language of your choice.

   Russian is selected by default.

10. Click **Save**.

After clicking **Save**, you are prompted to accept the terms and conditions of the End Use License Agreement. If you do not accept the End Use License Agreement, you cannot save the settings or continue to use the functionality.

After saving the settings, the available number of tokens is displayed. The tokens are replenished every day.

Integration is configured, you can proceed to the alerts and incidents analysis.

Analyzing using KIRA

After configuring the integration, you can analyze alerts and incidents using KIRA.

To perform an analysis:

1. Open the alert or incident details.
2. In the toolbar in the upper part of the alert or incident details, click **AI analysis** → **Analyze with KIRA**.

   If you have enough tokens, the analysis and the **Query to KIRA** task are started.

   Processing the request may take 30 seconds or longer.

   Tokens are expended even if the request returns an error saying that the requested topic is in the deny list; the information about remaining tokens is also updated.

The alert or incident is analyzed.

To view the analysis results, in the toolbar in the upper part of the alert or incident details, click **AI analysis** → **View report**. In the **Analyze with KIRA** window, the analysis result is displayed with the following details:

- A short summary of the alert or incident.
- The observables list.
- Alert or incident timeline with the event chronology.
- Case report.
- Date and time of the last request.
- Number of the expended tokens.

You can click the **Reanalyze** button to performs a new analysis disregarding the data of the previous analysis stored in the cache; the analysis expends tokens.

Possible errors of the Analyze using KIRA task

Errors may occur during alerts and incidents analysis using KIRA.

Possible errors

HTTP code	Description
400	Invalid request to the service or invalid client certificate. Possible reasons: required parameters are missing or contain invalid values; the client certificate used for KSN is expired or issued by an untrusted certification authority.
401	Authentication failed. The client certificate is valid, but there is no corresponding record in the limits database. Possible reasons: the record was deleted, was not created in time, or incorrect data was published due to a technical error.
403	Daily token limit has been exceeded.
404	Unknown or unsupported user task
413	Request token limit exceeded. The request contains too many input tokens. The maximum allowed number is 16,000 tokens per request, including service parameters. If this limit is exceeded, analysis is not performed.
500	Internal service error. An unspecified error occurred while processing the request.
502	AI service is unavailable. KIRA is not reachable.
503	Service connection error. Failed to obtain an access token from KIRA.
Other	Connection error. Any other error code returned by the KSN service that is not listed above.
No code	Request processing error on the KUMA side. Possible reasons include failure to read the response from KSN, failure to save data to the database, or failure to retrieve integration settings.

Creating search queries in natural language by using KIRA

In the **Threat hunting** section, you can enter a search query using natural language, automatically convert the query using Kaspersky Investigation & Response Assistant (KIRA) to the database query syntax, check the correctness of the query and refine it if necessary, and then run the query generated by the system and get event search results. Each time a request is sent, an audit event is generated. If an error occurs when generating a request, an error code is displayed.

The token limit is shared across the following options:

- Making search queries in natural language using KIRA
- Analyze with KIRA

The total amount of tokens depends on the query and the response. You can preview the number of tokens that will be spent on a query and the response, and edit the response to spend fewer tokens.

This functionality is available in the RU region if the following conditions are satisfied:

- There is a current license covering the AI module.
- A certificate was uploaded when configuring the KIRA integration. You can get the certificate file in PFX format, packed in the `<company name>`.ZIP archive, and the password for the certificate from Technical Support.
- The user has one of the following roles with corresponding access rights: General administrator, Administrator, Tier 2 analyst, Tier 1 analyst, or Junior analyst. Only a user with the General administrator role can configure the integration.

Creating search queries in natural language and converting them into SQL queries by using KIRA

Having configured the KIRA integration, you can start generating search queries to find events.

To generate a natural-language search query and convert it to an SQL query using KIRA:

1. In the **Threat hunting** section, below the SQL query field click the button ( ).
2. This opens the **KIRA SQL query generator** window; in that window, in the query field, enter the text of the query that you want KIRA to convert to SQL.

   For example: "Show all events from January 15 to 16, 2024, where the file name was "file.txt"".

   The conversion of queries in the Russian and English languages is supported.
3. Click the **Generate** button.

   The generated SQL query is displayed in the output field.

   For example:

   ```
   SELECT *
   ```

```
FROM events
WHERE FileName = 'file.txt'
ORDER BY Timestamp DESC
```

The generated time period is displayed in the **Time period** field:

For example:

2024-01-15T00:00:00.000 - 2024-01-16T00:00:00.000

If you have not specified the time range for your event search in your input, the period set on the **Events** tab is used. For example, "5 minutes" was set in **Events**, and the query was issued on January 15 at 15:30, the time period is set to 2026-01-15T15:25:00.000 - 2026-01-15T15:30:00.000

You can edit the values in the **SQL query** and **Time interval** fields before applying the generated query.

4. If you want to know how many tokens will be spent on sending the query, click **Calculate cost in tokens**.

5. If you want to use the generated SQL query to perform an event search, click the **Apply** button.

   The **KIRA SQL query generator** window is automatically closed.

   The generated SQL query is automatically inserted into the main query field on the **Threat hunting** tab.

   The absolute time range field is automatically populated with values from the generated query.

   For example: 2026-01-15T00:00:00.000 - 2026-01-15T00:00:00.000

6. If you want to perform an event search using the generated SQL query, click **Run query**.

The natural-language search query is converted to an SQL query and an event search result is obtained.

Possible errors while creating a query in natural language

Possible errors

HTTP code	Description
400	Invalid client certificate.
404	Error in request.
401	Missing certificate information. Contact Technical Support.
403	Daily limit of tokens exhausted.
413	Maximum number of tokens for request reached. Make the request smaller.
422	Natural-language request cannot be converted to SQL syntax.
500	Unknown error of the service.
502	KIRA service unavailable.
503	Error getting access token in service.
Other	Unknown error.
No code	Error while processing the request.

Configuring DLL Hijacking detection

DLL Hijacking is an attack technique that involves delivering vulnerable legitimate software along with a malicious dynamic link library (DLL) to the target system. When the vulnerable software is launched, it does not verify the legitimacy of the dynamic library and loads it by file name. As a result, malicious code is executed in the context of legitimate software. A DLL Hijacking attack is difficult to detect because the software being launched is legitimate. To detect such attacks, KUMA uses the AI module. The AI module analyzes the launch and runtime parameters of applications and identifies suspicious launches of legitimate software with malicious libraries.

To get a verdict, KUMA sends a request to KSN. The request contains information from the event fields that is needed for analysis. In response to a request, KSN sends one of the following analysis results:

- `0` – **Unclassified**. To get a result, you need to re-send the request. You can configure the re-sending of requests using the **Perform a repeat request setting**.
- `1` – **Unknown**. The library is not considered malicious at the time when the verdict is received.
- `2` – **Suspicious**. Getting such a result creates an alert if you have the corresponding correlation rule configured.
- `3` – **Malware**. Getting such a result creates an alert if you have the corresponding correlation rule configured. The **Malware** verdict represents a higher-likelihood detection of DLL Hijacking than the **Suspicious** verdict.

The result is written to the KL_AI_DLLHijackingCheckResult field of the event as a numerical value and a text comment.

DLL Hijacking attacks are detected at the event enrichment stage. This is achieved using the **check DLL Hijacking** enrichment type. You can embed an enrichment of this type in the collector or the correlator. We recommend embedding this type of enrichment in the correlator. In this way, the load on the KSN service is significantly lower than when checking events in the collector.

> To use an enrichment of the **check DLL Hijacking** type, your license must include the AI module. The General administrator must also accept the additional KSN license agreement in the **Kaspersky Security Network** section (**Settings** → **Integrations** → **AI services** → **Kaspersky Security Network**). If your instance of KUMA has multiple General administrators, it is sufficient for one of them to accept the agreement.

Detecting DLL Hijacking attacks using event enrichment in the correlator

To configure DLL Hijacking attack detection using event enrichment in the correlator:

1. In the application web interface, select **Resources** → **Correlators**.
2. Start creating a correlator or open the settings of an existing correlator.
3. Add an enrichment of the check DLL Hijacking type.

   When configuring the enrichment, you need to specify a list of fields corresponding to the parameters that are needed to create a KSN request. Make sure the fields you specify are populated as part of normalization in the collector.

4. In the enrichment settings, in the list under **Filter parameters**, select the following filter: **[OOTB] Events for DLLHijacking enrichment**. Filter for correlator. If this filter is missing from the list, import it from the repository. You can also configure the filter on your own.

5. If you want to configure the application to generate an alert when it gets a **Suspicious** or **Malware** verdict, create a correlation rule that does that. When creating the correlation rule, select the simple or standard type and on the **Selector** tab, specify two conditions in the filter parameters:

   `KL_AI_DLLHijackingCheckResult = 2` OR `KL_AI_DLLHijackingCheckResult = 3`

   Also, in the correlation rule, specify the fields that you specified when adding the enrichment (see step 3 of the instructions). The fields must be listed in the **Propagated fields** setting if you created a correlation rule of the simple type, or in the **Identical fields** setting if you created a correlation rule of the standard type.

Detecting DLL Hijacking attacks using event enrichment in the collector

> We do not recommend using the **DLL Hijacking** enrichment in the collector. This increases the load on the KSN service. Instead, add an enrichment of this type to the correlator.

To configure DLL Hijacking attack detection using event enrichment in the collector:

1. In the application web interface, select **Resources** → **Collectors**.

2. Start creating a collector or open the settings of an existing collector.

3. Add an enrichment of the check DLL Hijacking type.

   When configuring the enrichment, you need to specify a list of fields corresponding to the parameters that are needed to create a KSN request. Make sure the fields you specify are populated as part of normalization in the collector.

4. In the enrichment settings, in the list under **Filter parameters**, select the following filter: **[OOTB] Events for DLLHijacking enrichment**. Filter for collector. If this filter is missing from the list, import it from the repository. You can also configure the filter on your own.

5. If you want to configure the application to generate an alert when it gets a **Suspicious** or **Malware** verdict, create a correlation rule that does that. When creating the correlation rule, select the simple or standard type and on the **Selector** tab, specify two conditions in the filter parameters:

   `KL_AI_DLLHijackingCheckResult = 2` OR `KL_AI_DLLHijackingCheckResult = 3`

   Also, in the correlation rule, specify the fields that you specified when adding the enrichment (see step 3 of the instructions) as well as the KL_AI_DLLHijackingCheckResult field. The fields must be listed in the **Propagated fields** setting if you created a correlation rule of the simple type, or in the **Identical fields** setting if you created a correlation rule of the standard type.

AI asset score and status

The **AI asset score and status** service can be installed if your license covers the AI module.

The AI service helps with precisely assessing the severity of correlation events generated by triggered correlation rules.

The AI service gets correlation events with a non-empty **Affected assets** field from the available storage clusters, constructs the expected sequence of events, and trains the AI model. Based on the chain of triggered correlation rules, the AI service calculates whether such a sequence of events is typical for this infrastructure. Non-typical patterns increase the score of the asset.

The AI service calculates the **Score AI** and the **Status AI**, which are displayed in the asset card. If you remove the license key, the **Score AI** and **Status AI** fields are hidden from the asset card. If you add the license key again, the values of the **Score AI** and **Status AI** fields are shown again.

The score is a number that quantifies how non-typical the activity on the asset is, and whether it is worth paying attention to. Possible values of the **Status AI** field: **Low**, **Medium**, **High**, **Critical**. The score is a number in the range from 0 to 1.

There are four ranges that correspond to statuses:

Low: 0 ≤ score < 0.25

Medium: 0.25 ≤ score < 0.5

High: 0.5 ≤ score < 0.75

Critical: 0.75 ≤ score ≤ 1

You can apply a filter by the **Score AI** and **Status AI** fields when searching for assets. You can also set up proactive categorization of assets by the **Score AI** and **Status AI** fields, which moves the asset to the category corresponding to the risk level as soon as the AI service assigns a score to the asset.

You can create a structure of multiple categories and automatically populate these with assets in accordance with the calculated risk values.

In the **Settings** → **Asset audit** section, you can configure audit events to be generated when an asset is added to a category. Audit events can be taken into account in correlation rules, and you can monitor them on the dashboard and in reports.

To monitor asset category changes on the dashboard, create an **Events** widget with a query similar to the following:

```
SELECT count(ID) AS `metric`,
formatDateTime(toTimeZone(fromUnixTimestamp64Milli(Timestamp), 'Europe/Moscow'),
'%d.%m.%Y %H:%m:%S') AS `value` FROM `events`

where DeviceVendor = 'Kaspersky' and DeviceProduct = 'KUMA' and

DeviceEventCategory = 'Audit assets' and DeviceAction= 'asset added to category'

and DeviceCustomString1 = 'Main/Categorized assets/ML/score>0.5'
```

```
GROUP BY Timestamp ORDER BY value LIMIT 250
```

To monitor the distribution of assets by status on the dashboard, create an **Assets by severity** widget. The **Assets by severity** widget is available if the license includes the AI module. The pie chart indicates the numbers of assets grouped by status.

Every time the AI service is restarted, the AI service trains the model from scratch and reassesses the score of the assets mentioned in events of the current day.

The directory specified in the configuration file stores events that the AI service got from KUMA storage clusters for the specified number of days. For example, if the configuration file specifies 12 days, the AI service gets events for the past 12 days. The oldest events are deleted from the directory. The trained model is stored in the same directory.

The model is retrained at midnight UTC. The asset score is reassessed once an hour for all assets that were mentioned in events of the current day (UTC).

Service logs are stored in /var/log/syslog.

Installing the AI asset score and status service

Installing the AI asset score and status service

To install the service:

1. Extract the **mlservice-installer-XX.tgz** archive that is included in the distribution kit.

   The **mlservice-installer-XX.tgz** archive contains scripts for installing and removing the service, as well as the config.yaml configuration file.

2. In the config.yaml configuration file, in the `kuma_address` setting, specify the FQDN of the host on which KUMA Core is installed and the port on which the KUMA Core is to listen for AI service connections.

   In a high availability configuration, you must specify port 7226. You can keep default values for the rest of the settings. After installation, the service starts with the settings specified in the config.yaml file.

3. If you want to install the service on a remote host, specify the address of the remote host in the inventory.yaml file and make sure you have network access. By default, the service is installed on the local host as specified in inventory.yaml.

4. Get a Core certificate in the KUMA web interface. To do this, in the main window of the KUMA web interface, click the name of your user account in the lower part of window to open a menu and select **AI service certificate**. The certificate is downloaded to your default download directory.

   If necessary, you can reissue the AI service certificate.

5. Rename the downloaded certificate file to **core-external-ca.cert**.

6. Save the KUMA Core certificate file in the roles/mlservice/files directory under the installer directory.

7. Change to the directory with the service files and from that directory, run the following command:

   ```
   ./install < path to inventory.yaml >
   ```

8. If you accept the terms and conditions of the End User License Agreement, press `Y`. If you do not accept the terms and conditions of the End User License Agreement, you cannot proceed with the installation of the service. You can find the file with the text of the End User License Agreement in the mlservice-installer/eula directory.

9. The installer generates a certificate and a key during the installation process and places these in the directories specified in the config.yaml configuration file. You must upload the certificate to KUMA.

   In the KUMA web interface, in the **Settings** → **Integrations** → **AI services** section, in the **AI asset score and status** window, fill in the following fields:

   a. In the **URL** field, specify the FQDN of the KUMA Core host on and the port on which the KUMA Core listens for the AI service, for example: <FQDN of the KUMA Core host>:7226. The port number must match the port number specified in the configuration file. Make sure the port is not being used by other applications.

      For a KUMA installation in high-availability configuration, the **URL** field is not displayed in the interface, the port value is taken from the KUMA_APPRAISER_AI_API_PORT environment variable and the port is opened for all IP addresses of the KUMA Core host.

b. In the **Certificate** drop-down list, select **Create new** to open the **Create secret** window; in that window, specify **Certificate** as the secret type and upload the certificate from the directory specified in the config.yaml configuration file.

c. Enable the **State** toggle switch. By default, the toggle switch is disabled.

d. Click **Save**.

Immediately after installation, the service will make attempts to connect to KUMA for 15 minutes with 1-minute intervals. If no certificate is added in the KUMA web interface, the connection will fail and the service will stop. In this case, you can add a certificate and restart the AI service; the service makes new attempts to connect.

After saving the settings, get the log of the Core server and make sure it does not contain the "<port number>: bind: address already in use" error

The AI service is installed.

Removing the AI asset score and status service

To remove the AI service, change to the directory with the AI service files, and from that directory, run the following:

```
./uninstall <path to inventory.yaml>
```

Reissuing the AI service certificate

To reissue the AI service certificate:

1. In the KUMA web interface, go to the **Settings** → **Other** → **Common** section.
2. Under **Core settings**, for the **AI service certificate** setting, click the **Reissue** button.
3. Click **OK** to confirm the operation.
4. Replace the certificate in the /opt/kaspersky/mlservice directory with the new certificate.
5. Restart the Score AI and Status AI service by running the command `systemctl restart mlservice`; also restart the KUMA Core.

The new AI service certificate is applied.

Settings of the AI asset score and status service

Available AI service settings

Setting	Description
`cert_file_path`	Path to the directory that contains the certificate generated by the installer. Default path: /opt/kaspersky/mlservice/service.crt You can specify a different path. In this case, make sure that the user that starts the service has access to the specified directory.
`key_file_path`	Path to the directory that contains the key generated by the installer. Default path: /opt/kaspersky/mlservice/service.key You can specify a different path. In this case, make sure that the user that starts the service has access to the specified directory.
`kuma_address`	FQDN of the host on which the KUMA Core is installed and port on which the KUMA Core is to listen for the AI service. To install in a high availability configuration, you must specify port 7226. Example: <FQDN of the host on which KUMA Core is installed>:7226
`kuma_cert_file_path`	Path to the directory where the KUMA Core certificate is located. Default path: /opt/kaspersky/mlservice/core-external-ca.cert
`event_storage_path`	Path to the directory where the service is to place received correlation events. Default path: /var/mlservice/events
`model_storage_path`	Path to the directory where the service is to place the trained model. Default path: /var/mlservice/models
`period_for_train_days`	The number of days for which you want to get correlation events that involve your assets from the available storage clusters in order to train the model. The default setting is 12 days. This means that the directory will contain events for the past <N> days. The oldest events are deleted.
`events_overlap_in_seconds`	Overlap time. When events for assessing the scores of assets are downloaded on a schedule, they are retrieved from the time of the last downloaded event for the current day minus the value of the events_overlap_in_seconds setting. The default value is 60 seconds. Example: the time when the last event was received is 8:58. The starting time for the next batch of events to be downloaded is 8:57.

Configuring event sources

This section provides information on configuring the receipt of events from various sources.

Configuring receipt of Auditd events

KUMA lets you monitor and audit the Auditd events on Linux devices.

Before configuring event receiving, make sure to create a new KUMA collector for the Auditd events.

Configuring the receipt of Auditd events involves the following steps:

1. Installation of KUMA collector in the network infrastructure.
2. Configuring the event source server.
3. Verifying receipt of Auditd events by the KUMA collector.

   You can verify that the Auditd event source server is configured correctly by searching for related events in the KUMA Console.

Installing KUMA collector for receiving Auditd events

After creating a collector, in order to configure event receiving using rsyslog, you must install a collector on the network infrastructure server intended for receiving events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring the event source server

The rsyslog service is used to transmit events from the server to the KUMA collector.

To configure transmission of events from the server to the collector:

1. Make sure that the rsyslog service is installed on the event source server. For this purpose, execute the following command:

   ```
   systemctl status rsyslog.service
   ```

   If the rsyslog service is not installed on the server, install it by executing the following command:

   - for RPM-based distributions:

     ```
     yum install rsyslog
     ```

   - for Debian or Ubuntu-based distributions:

     ```
     apt-get install rsyslog
     ```

   Enable the service by executing the following command:

   ```
   systemctl enable rsyslog.service
   systemctl start rsyslog.service
   ```

2. In the /etc/rsyslog.d folder, create the audit.conf file with the following content:

   ```
   $ModLoad imfile
   ```

```
$InputFileName /var/log/audit/audit.log
$InputFileTag tag_audit_log:
$InputFileStateFile audit_log
$InputFileSeverity info
$InputFileFacility local6
$InputRunFileMonitor
*.* @<KUMA collector IP address>:<KUMA collector port>
```

> If you want to send events over TCP, instead of the last line in the file insert the following line:
> `*.* @@<KUMA collector IP address>:<KUMA collector port>`.

3. Save the changes to the audit.conf file.

4. Restart the rsyslog service by executing the following command:

   ```
   systemctl restart rsyslog.service
   ```

The event source server is configured. Data about events is transmitted from the server to the KUMA collector.

Configuring receipt of KATA/EDR events

You can configure the receipt of Kaspersky Anti Targeted Attack Platform events in the KUMA SIEM system.

Before configuring event receipt, make sure to create a KUMA collector for the KATA/EDR events.

> When creating a collector in the KUMA Console, make sure that the port number matches the port specified in step 4c of Configuring export of Kaspersky Anti Targeted Attack Platform events to KUMA, and that the connector type corresponds to the type specified in step 4d.

> To receive Kaspersky Anti Targeted Attack Platform events using Syslog, in the collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KATA** normalizer.

Configuring the receipt of KATA/EDR events involves the following steps:

1. Configuring the forwarding of KATA/EDR events

2. Installing the KUMA collector in the network infrastructure

3. Verifying receipt of KATA/EDR events in the KUMA collector

   You can verify that the KATA/EDR event source server is configured correctly by searching for related events in the KUMA Console. Kaspersky Anti Targeted Attack Platform events are displayed as KATA in the table with search results.

Configuring export of KATA/EDR events to KUMA

To configure export of events from Kaspersky Anti Targeted Attack Platform to KUMA:

1. In a browser on any computer with access to the Central Node server, enter the IP address of the server hosting the Central Node component.

   A window for entering Kaspersky Anti Targeted Attack Platform user credentials opens.

2. In the user credentials entry window, select the **Local administrator** check box and enter the Administrator credentials.

3. Go to the **Settings** → **SIEM system** section.

4. Specify the following settings:

   a. Select the **Activity log** and **Detections** check boxes.

   b. In the **Host/IP** field, enter the IP address or host name of the KUMA collector.

   c. In the **Port** field, specify the port number to connect to the KUMA collector.

   d. In the **Protocol** field, select **TCP** or **UDP** from the list.

   e. In the **Host ID** field, specify the server host ID to be indicated in the SIEM systems log as a detection source.

   f. In the **Alert frequency** field, enter the interval for sending messages: from 1 to 59 minutes.

   g. Enable TLS encryption, if necessary.

   h. Click **Apply**.

Export of Kaspersky Anti Targeted Attack Platform events to KUMA is configured.

Creating KUMA collector for receiving KATA/EDR events

After configuring the event export settings, you must create a collector for Kaspersky Anti Targeted Attack Platform events in the KUMA Console.

For details on creating a KUMA collector, refer to Creating a collector.

When creating a collector in the KUMA Console, make sure that the port number matches the port specified in step 4c of Configuring export of Kaspersky Anti Targeted Attack Platform events to KUMA, and that the connector type corresponds to the type specified in step 4d.

To receive Kaspersky Anti Targeted Attack Platform events using Syslog, in the collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KATA** normalizer.

Installing KUMA collector for receiving KATA/EDR events

After creating a collector, to configure receiving Kaspersky Anti Targeted Attack Platform events, install a new collector on the network infrastructure server intended for receiving events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring Open Single Management Platform for export of events to the KUMA SIEM-system

KUMA allows you to receive and export events from Open Single Management Platform Administration Server to the KUMA SIEM system.

Configuring the export and receipt of Open Single Management Platform events proceeds in stages:

1. Configuring the export of Open Single Management Platform events.
2. Configuring the KUMA Collector.
3. Installing the KUMA collector in the network infrastructure.
4. Verifying receipt of Open Single Management Platform events in the KUMA collector

   You can verify if the events from Open Single Management Platform Administration Server were correctly exported to the KUMA SIEM system by using the KUMA Console to search for related events.

   To display Open Single Management Platform events in CEF format in the table, enter the following search expression:

   ```
   SELECT * FROM `events` WHERE DeviceProduct = 'KSC' ORDER BY Timestamp DESC LIMIT
   250
   ```

Configuring KUMA collector for collecting Open Single Management Platform events

After configuring the export of events in the CEF format from Open Single Management Platform Administration Server, configure the collector in the KUMA Console.

To configure the KUMA Collector for Open Single Management Platform events:

1. In the KUMA Console, select **Resources** → **Collectors**.
2. In the list of collectors, find the collector with the **[OOTB] KSC** normalizer and open it for editing.

3. At the **Transport** step, in the **URL** field, specify the port to be used by the collector to receive Open Single Management Platform events.

   The port must match the port of the KUMA SIEM system server.

4. At the **Event parsing** step, make sure that the **[OOTB] KSC** normalizer is selected.

5. At the **Routing** step, make sure that the following destinations are added to the collector resource set:

   - **Storage**. To send processed events to the storage.
   - **Correlator**. To send processed events to the correlator.

   If the **Storage** and **Correlator** destinations were not added, create them.

6. At the **Setup validation** tab, click **Create and save service**.

7. Copy the command for installing the KUMA collector that appears.

Installing KUMA collector for collecting Open Single Management Platform events

After configuring the collector for collecting Open Single Management Platform events in the CEF format, install the KUMA collector on the network infrastructure server intended for receiving events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring receiving Open Single Management Platform event from MS SQL

KUMA allows you to receive information about Open Single Management Platform events from an MS SQL database.

Before configuring, make sure that you have created the KUMA collector for Open Single Management Platform events from MS SQL.

> When creating the collector in the KUMA web interface, at the **Transport** step, select the **[OOTB] KSC SQL** connector.

> To receive Open Single Management Platform events from the MS SQL database, at the **Event parsing** step, select the **[OOTB] KSC from SQL** normalizer.

Configuring event receiving consists of the following steps:

1. Creating an account in the MS SQL.
2. Configuring the SQL Server Browser service.
3. Creating a secret.

4. Configuring a connector.

5. Installing a collector in the network infrastructure.

6. Verifying receipt of events from MS SQL in the KUMA collector.

   You can verify that the receipt of events from MS SQL is configured correctly by searching for related events in the KUMA web interface.

Creating an account in the MS SQL database

To receive Kaspersky Security Center events from MS SQL, a user account is required that has the rights necessary to connect and work with the database.

To create an account for working with MS SQL:

1. Log in to the server with MS SQL for Kaspersky Security Center installed.

2. Using **SQL Server Management Studio**, connect to MS SQL using an account with administrator rights.

3. In the Object Explorer pane, expand the **Security** section.

4. Right-click the **Logins** folder and select **New Login** from the context menu.

   The **Login - New** window opens.

5. On the **General** tab, click the **Search** button next to the **Login name** field.

   The **Select User or Group** window opens.

6. In the **Enter the object name to select (examples)** field, specify the object name and click **OK**.

   The **Select User or Group** window closes.

7. In the **Login - New** window, on the **General** tab, select the **Windows authentication** option.

8. In the **Default database** field, select the Kaspersky Security Center database.

   The default Kaspersky Security Center database name is KAV.

9. On the **User Mapping** tab, configure the account permissions:

   a. In the **Users mapped to this login** section, select the Kaspersky Security Center database.

   b. In the **Database role membership for** section, select the check boxes next to the **db_datareader** and **public** permissions.

10. On the **Status** tab, configure the permissions for connecting the account to the database:

    - In the **Permission to connect to database engine** section, select **Grant**.

    - In the **Login** section, select **Enabled**.

11. Click **OK**.

    The **Login - New** window closes.

To check the account permissions:

1. Run **SQL Server Management Studio** using the created account.
2. Go to any MS SQL database table and make a selection based on the table.

Configuring the SQL Server Browser service

After creating an account in MS SQL, you must configure the SQL Server Browser service.

To configure the SQL Server Browser service:

1. Open **SQL Server Configuration Manager**.
2. In the left pane, select **SQL Server Services**.
   A list of services opens.
3. Open the **SQL Server Browser** service properties in one of the following ways:
   - Double-click the name of the **SQL Server Browser** service.
   - Right-click the name of the **SQL Server Browser** service and select **Properties** from the context menu.
4. In the **SQL Server Browser Properties** window that opens, select the **Service** tab.
5. In the **Start Mode** field, select **Automatic**.
6. Select the **Log On** tab and click the **Start** button.
   Automatic startup of the **SQL Server Browser** service is enabled.
7. Enable and configure the **TCP/IP** protocol by doing the following:
   a. In the left pane, expand the **SQL Server Network Configuration** section and select the **Protocols for <SQL Server name>** subsection.
   b. Right-click the **TCP/IP** protocol and select **Enable** from the context menu.
   c. In the **Warning** window that opens, click **OK**.
   d. Open the **TCP/IP** protocol properties in one of the following ways:
      - Double-click the **TCP/IP** protocol.
      - Right-click the **TCP/IP** protocol and select **Properties** from the context menu.
   e. Select the **IP Addresses** tab, and then in the **IPALL** section, specify port 1433 in the **TCP Port** field.
   f. Click **Apply** to save the changes.
   g. Click **OK** to close the window.

8. Restart the **SQL Server (<SQL Server name>)** service by doing the following:

   a. In the left pane, select **SQL Server Services**.

   b. In the service list on the right, right-click the **SQL Server (<SQL Server name>)** service and select **Restart** from the context menu.

9. In **Windows Defender Firewall with Advanced Security**, allow inbound connections on the server on the TCP port 1433.

Creating a secret in KUMA

After creating and configuring an account in MS SQL, you must add a secret in the KUMA Console. This resource is used to store credentials for connecting to MS SQL.

To create a KUMA secret:

1. In the KUMA Console, open the **Resources** → **Secrets** section.

   The list of available secrets will be displayed.

2. Click the **Add secret** button to create a new secret.

   The secret window is displayed.

3. Enter information about the secret:

   a. In the **Name** field, choose a name for the added secret.

   b. In the **Tenant** drop-down list, select the tenant that will own the created resource.

   c. In the **Type** drop-down list, select **urls**.

   d. In the **URL** field, specify a string of the form:

   `sqlserver://[<domain>%5C]<username>:<password>@<server>:1433/<database_name>`

   where:

   - `domain` is a domain name.
   - `%5C` is the domain/user separator. Represents the "\" character in URL format.
   - `username` is the name of the created MS SQL account.
   - `password` is the password of the created MS SQL account.
   - `server` is the name or IP address of the server where the MS SQL database for Kaspersky Security Center is installed.
   - `database_name` is the name of the Kaspersky Security Center database. The default name is KAV.

   Example:
   `sqlserver://test.local%5Cuser:password123@10.0.0.1:1433/KAV`

If the MS SQL database account password contains special characters (@ # $ % & * ! + = [ ] : ' , ? / \ ` ( ) ;), convert them to URL format.

4. Click **Save**.

For security reasons, the string specified in the URL field is hidden after the secret is saved.

Configuring a connector

To connect KUMA to an MS SQL database, you must configure the connector.

To configure a connector:

1. In the KUMA Console, go to the **Resources** → **Connectors** section.
2. In the list of connectors, find the **[OOTB] KSC SQL** connector and open it for editing.

   If a connector is not available for editing, copy it and open the connector copy for editing.

   If the **[OOTB] KSC SQL** connector is not available, contact your system administrator.

3. On the **Basic settings** tab, in the **URL** drop-down lists, select the secret created for connecting to the MS SQL database.
4. Click **Save**.

Configuring the KUMA Collector for receiving Kaspersky Security Center events from an MS SQL database

After configuring the event export settings, you must create a collector in the KUMA Console for Kaspersky Security Center events received from MS SQL.

For details on creating a KUMA collector, refer to Creating a collector.

When creating the collector in the KUMA Console, at the **Transport** step, select the **[OOTB] KSC SQL connector**.

To receive Kaspersky Security Center events from MS SQL, at the **Event parsing** step, select the **[OOTB] KSC from SQL** normalizer.

Installing the KUMA Collector for receiving Kaspersky Security Center events from the MS SQL database

After configuring the collector for receiving Kaspersky Security Center events from MS SQL, install the KUMA collector on the network infrastructure server where you intend to receive events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring receipt of events from Windows devices using KUMA Agent (WEC)

KUMA allows you to receive information about events from Windows devices using the WEC KUMA Agent.

Configuring event receiving consists of the following steps:

1. Configuring policies for receiving events from Windows devices.
2. Configuring centralized receipt of events using the Windows Event Collector service.
3. Granting permissions to view events.
4. Granting permissions to log on as a service.
5. Configuring the KUMA Collector.
6. Installing KUMA collector.
7. Forwarding events from Windows devices to KUMA.

Configuring audit of events from Windows devices

You can configure event audit on Windows devices for an individual device or for all devices in a domain.

This section describes how to configure an audit on an individual device and how to use a domain group policy to configure an audit.

Configuring an audit policy on a Windows device

To configure audit policies on a device:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.

3. Select **Security Settings** → **Local policies** → **Audit policy**.

4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

Configuration of an audit policy on the device is complete.

Configuring an audit using a group policy

In addition to configuring an audit policy on an individual device, you can also configure an audit by using a domain group policy.

To configure an audit using a group policy:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.

3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **Audit policy**.

4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events

- Audit Logon Events
- Audit Account Management

> If you want to receive Windows logs from a large number of servers or if installation of KUMA agents on domain controllers is not allowed, it is recommended to configure Windows log redirection to individual servers that have the Windows Event Collector service configured.

The audit policy is now configured on the server or workstation.

Configuring centralized receipt of events from Windows devices using the Windows Event Collector service

The Windows Event Collector service allows you to centrally receive data about events on servers and workstations running Windows. You can use the Windows Event Collector service to subscribe to events that are registered on remote devices.

You can configure the following types of event subscriptions:

- **Source-initiated subscriptions**. Remote devices send event data to the Windows Event Collector server whose address is specified in the group policy. For details on the subscription configuration procedure, please refer to the Configuring data transfer from the event source server section.
- **Collector-initiated subscriptions**. The Windows Event Collector server connects to remote devices and independently gathers events from local logs. For details on the subscription configuration procedure, please refer to the Configuring the Windows Event Collector service section.

Configuring data transfer from the event source server

You can receive information about events on servers and workstations by configuring data transfer from remote devices to the Windows Event Collector server.

Preliminary steps

1. Verify that the Windows Remote Management service is configured on the event source server by running the following command in the PowerShell console:

   ```
   winrm get winrm/config
   ```

   If the Windows Remote Management service is not configured, initialize it by running the following command:

   ```
   winrm quickconfig
   ```

2. If the event source server is a domain controller, make the Windows logs available over the network by running the following command in PowerShell as an administrator:

   ```
   wevtutil set-log security /ca:’O:BAG:SYD:(A;;0xf0005;;;SY)(A;;0x5;;;BA)(A;;0x1;;;S-1-5-32-573)(A;;0x1;;;S-1-5-20)
   ```

   Verify access by running the following command:

   ```
   wevtutil get-log security
   ```

Configuring the firewall on the event source server

To enable the Windows Event Collector server to receive Windows log entries, inbound connection ports must be opened on the event source server.

To open ports for inbound connections:

1. On the event source server, open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `wf.msc` and click **OK**.

   The **Windows Defender Firewall with Advanced Security** window opens.
3. Go to the **Inbound Rules** section and click **New Rule** in the **Actions** pane.

   The **New Inbound Rule Wizard** opens.
4. At the **Rule type** step, select **Port**.
5. At the **Protocols and ports** step, select **TCP** as the protocol. In the **Specific local ports** field, indicate the relevant port numbers:

   - `5985` (for HTTP access)
   - `5986` (for HTTPS access)

   You can indicate one of the ports, or both.
6. At the **Action** step, select **Allow connection** (selected by default).
7. At the **Profile** step, clear the **Private** and **Public** check boxes.
8. At the **Name** step, specify a name for the new inbound connection rule and click **Done**.

Configuration of data transfer from the event source server is complete.

> The Windows Event Collector server must have the permissions to read Windows logs on the event source server. These permissions can be assigned to both the Windows Event Collector server account and to a special user account. For details on granting permissions, please refer to the Granting user permissions to view the Windows Event Log.

Configuring the Windows Event Collector service

The Windows Event Collector server can independently connect to devices and gather data on events of any severity.

To configure the receipt of event data by the Windows Event Collector server:

1. On the event source server, open the **Run** window by pressing **`Win`**+**`R`**.
2. In the opened window, type `services.msc` and click **OK**.

   The **Services** window opens.
3. In the list of services, find and start the **Windows Event Collector** service.

4. Open the **Event Viewer** snap-in by doing the following:

   a. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

   b. In the opened window, type `eventvwr` and click **OK**.

5. Go to the **Subscriptions** section and click **Create Subscription** in the **Actions** pane.

6. In the opened **Subscription Properties** window, specify the name and description of the subscription, and define the following settings:

   a. In the **Destination log** field, select **Forwarded events** from the list.

   b. In the **Subscription type and source computers** section, click the **Select computers** button.

   c. In the opened **Computers** window, click the **Add domain computer** button.

      The **Select computer** window opens.

   d. In the **Enter the object names to select (examples)** field, list the names of the devices from which you want to receive event information. Click **OK**.

   e. In the **Computers** window, check the list of devices from which the Windows Event Collector server will gather event data and click **OK**.

   f. In the **Subscription properties** window, in the **Collected events** field, click the **Select events** button.

   g. In the opened **Request filter** window, specify how often and which data about events on devices you want to receive.

   h. If necessary, in the **<All event codes>** field, list the codes of the events whose information you want to receive or do not want to receive. Click **OK**.

7. If you want to use a special account to view event data, do the following:

   a. In the **Subscription properties** window, click the **Advanced** button.

   b. In the opened **Advanced subscription settings** window, in the user account settings, select **Specific user**.

   c. Click the **User and password** button and enter the account credentials of the selected user.

Configuration of the Event Collector Service is complete.

To verify that the configuration is correct and event data is being received by the Windows Event Collector server:

In the **Event Viewer** snap-in, go to **Event Viewer (Local)** → **Windows logs** → **Forwarded events**.

Granting permissions to view Windows events

You can grant permissions to view Windows events for a specific device or for all devices in a domain.

To grant permissions to view events on a specific device:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `compmgmt.msc` and click **OK**.

   The **Computer Management** window opens.
3. Go to **Computer Management (local)** → **Local users and groups** → **Groups**.
4. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.
5. Click the **Add** button at the bottom of the **Properties: Event Log Readers** window.

   The **Select Users, Computers or Groups** window opens.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

To grant permissions to view events for all devices in a domain:

1. Log in to the domain controller with administrator privileges.
2. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
3. In the opened window, type `dsa.msc` and click **OK**.

   The **Active Directory Users and Computers** window opens.
4. Go to **Active Directory Users and Computers** → **<Domain name>** → **Builtin**.
5. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

   In the **Properties: Event Log Readers** window, open the **Members** tab and click the **Add** button.

   The **Select Users, Computers or Groups** window opens.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

Granting permissions to log on as a service

You can grant permission to log on as a service to a specific device or to all devices in a domain. The **Log on as a service** permission allows you to start a process using an account that has been granted this permission.

*To grant the **Log on as a service** permission to a device:*

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.
3. Go to **Security settings** → **Local policies** → **User rights assignment**.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.

5. In the opened **Properties: Log on as a Service** window, click the **Add User or Group** button.

   The **Select Users or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the accounts or devices to which you want to grant the permission to log on as a service. Click **OK**.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

*To grant the **Log on as a service** permission to devices in a domain:*

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.

3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **User rights assignment**.

4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.

5. In the opened **Properties: Log on as a Service** window, click the **Add User or Group** button.

   The **Select Users or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant the permission to log on as a service. Click **OK**.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

Configuring the KUMA Collector for receiving events from Windows devices

After you finish configuring the audit policy on devices, creating subscriptions to events and granting all the necessary permissions, you need to create a collector in the KUMA Console for events from Windows devices.

For details on creating a KUMA collector, refer to Creating a collector.

To receive events from Windows devices, define the following collector settings in the KUMA Collector Installation Wizard:

1. At the **Transport** step, define the following settings:

   a. In the **Connector** window, select **Create**.

   b. In the **Type** field, select **http**.

   c. In the **Delimiter** field, select **\0**.

2. On the **Advanced settings** tab, in the **TLS mode** field, select **With verification**.

3. At the **Event parsing** step, click the **Add event parsing** button.

4. In the opened **Basic event parsing** window, in the **Normalizer** field, select **[OOTB] Windows Extended v.1.0** and click **OK**.

5. At the **Routing** step, add the following destinations:

   - **Storage**. To send processed events to the storage.

   - **Correlator**. To send processed events to the correlator.

   If the **Storage** and **Correlator** destinations were not added, create them.

6. At the **Setup validation** tab, click **Create and save service**.

7. Copy the command for installing the KUMA collector that appears.

Installing the KUMA Collector for receiving events from Windows devices

After configuring the collector for receiving Windows events, install the KUMA Collector on the server of the network infrastructure intended for receiving events.

For details on installing the KUMA collector, refer to the Installing collector in the network infrastructure section.

Configuring forwarding of events from Windows devices to KUMA using KUMA Agent (WEC)

To complete the data forwarding configuration, you must create a WEC KUMA agent and then install it on the device from which you want to receive event information.

For more details on creating and installing a WEC KUMA Agent on Windows devices, please refer to the Forwarding events from Windows devices to KUMA section.

Configuring receipt of events from Windows devices using KUMA Agent (WMI)

KUMA allows you to receive information about events from Windows devices using the WMI KUMA Agent.

Configuring event receiving consists of the following steps:

1. Configuring audit settings for managing KUMA.

2. Configuring data transfer from the event source server.

3. Granting permissions to view events.

4. Granting permissions to log on as a service.

5. Creating a KUMA collector.

   To receive events from Windows devices, in the KUMA Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Windows Extended v.1.0**.

6. Installing KUMA collector.

7. Forwarding events from Windows devices to KUMA.

   To complete the data forwarding configuration, you must create a WMI KUMA agent and then install it on the device from which you want to receive event information.

Configuring audit settings for managing KUMA

You can configure event audit on Windows devices both on a specific device using a local policy or on all devices in a domain using a group policy.

This section describes how to configure an audit on an individual device and how to use a domain group policy to configure an audit.

Configuring an audit using a local policy

To configure an audit using a local policy:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.

3. Select **Security Settings** → **Local policies** → **Audit policy**.

4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.

5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

Configuration of an audit policy on the device is complete.

Configuring an audit using a group policy

In addition to configuring an audit on an individual device, you can also configure an audit by using a domain group policy.

To configure an audit using a group policy:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **Audit policy**.
4. In the pane on the right, double-click to open the properties of the policy for which you want to enable an audit of successful and unsuccessful attempts.
5. In the **<Policy name> properties** window, on the **Local security setting** tab, select the **Success** and **Failure** check boxes to track successful and interrupted attempts.

   It is recommended to enable an audit of successful and unsuccessful attempts for the following policies:

   - Audit Logon
   - Audit Policy Change
   - Audit System Events
   - Audit Logon Events
   - Audit Account Management

The audit policy is now configured on the server or workstation.

Configuring data transfer from the event source server

Preliminary steps

1. On the event source server, open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `services.msc` and click **OK**.

   The **Services** window opens.
3. In the list of services, find the following services:

   - Remote Procedure Call
   - RPC Endpoint Mapper
4. Check the **Status** column to confirm that these services have the **Running** status.

Configuring the firewall on the event source server

The Windows Management Instrumentation server can receive Windows log entries if ports are open for inbound connections on the event source server.

To open ports for inbound connections:

1. On the event source server, open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `wf.msc` and click **OK**.

   The **Windows Defender Firewall with Advanced Security** window opens.
3. In the **Windows Defender Firewall with Advanced Security** window, go to the **Inbound Rules** section and in the **Actions** pane, click **New Rule**.

   This opens the **New Inbound Rule Wizard**.
4. In the **New Inbound Rule Wizard,** at the **Rule Type** step, select **Port**.
5. At the **Protocols and ports** step, select **TCP** as the protocol. In the **Specific local ports** field, indicate the relevant port numbers:
   - `135`
   - `445`
   - `49152-65535`
6. At the **Action** step, select **Allow connection** (selected by default).
7. At the **Profile** step, clear the **Private** and **Public** check boxes.
8. At the **Name** step, specify a name for the new inbound connection rule and click **Done**.

Configuration of data transfer from the event source server is complete.

Granting permissions to view Windows events

You can grant permissions to view Windows events for a specific device or for all devices in a domain.

To grant permissions to view events on a specific device:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `compmgmt.msc` and click **OK**.

   The **Computer Management** window opens.
3. Go to **Computer Management (local)** → **Local users and groups** → **Groups**.
4. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

5. Click the **Add** button at the bottom of the **Properties: Event Log Readers** window.

   The **Select Users, Computers or Groups** window opens.

6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

To grant permissions to view events for all devices in a domain:

1. Log in to the domain controller with administrator privileges.

2. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

3. In the opened window, type `dsa.msc` and click **OK**.

   The **Active Directory Users and Computers** window opens.

4. In the **Active Directory Users and Computers** window, go to the **Active Directory Users and Computers** section → **<Domain name>** → **Builtin**.

5. In the pane on the right, select the **Event Log Readers** group and double-click to open the policy properties.

   In the **Properties: Event Log Readers** window, open the **Members** tab and click the **Add** button.

   The **Select Users, Computers or Groups** window opens.

6. In the **Select User, Computer, or Group** window, In the **Enter the object name to select (examples)** field, list the names of the users or devices to which you want to grant permissions to view event data. Click **OK**.

Granting permissions to log on as a service

You can grant permission to log on as a service to a specific device or to all devices in a domain. The **Log on as a service** permission allows you to start a process using an account that has been granted this permission.

Before granting the permission, make sure that the accounts or devices to which you want to grant the **Log on as a service** permission are not listed in the properties of the **Deny log on as a service** policy.

To grant the Log on as a service permission to a device:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.

2. In the opened window, type `secpol.msc` and click **OK**.

   The **Local security policy** window opens.

3. In the **Local Security Policy** window, go to the **Security Settings** → **Local Policies** → **User Rights Assignment** section.

4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.

5. This opens the **Properties: Log on as a Service** window; in that window, click **Add User or Group**.

   This opens the **Select Users or Groups** window.

6. In the **Enter the object names to select (examples)** field, list the names of the accounts or devices to which you want to grant the permission to log on as a service. Click **OK**.

To grant the Log on as a service permission to devices in a domain:

1. Open the **Run** window by pressing the key combination **`Win`**+**`R`**.
2. In the opened window, type `gpedit.msc` and click **OK**.

   The **Local Group Policy Editor** window opens.
3. Select **Computer configuration** → **Windows configuration** → **Security settings** → **Local policies** → **User rights assignment**.
4. In the pane on the right, double-click to open the properties of the **Log on as a service** policy.
5. This opens the **Properties: Log on as a Service** window; in that window, click **Add User or Group**.

   This opens the **Select Users or Groups** window.
6. In the **Enter the object names to select (examples)** field, list the names of the users or devices to which you want to grant the permission to log on as a service. Click **OK**.

Configuring receipt of PostgreSQL events

KUMA lets you monitor and audit PostgreSQL events on Linux devices using rsyslog.

Events are audited using the pgAudit plugin. The plugin supports PostgreSQL 9.5 and later. For details about the pgAudit plugin, see https://github.com/pgaudit/pgaudit.

Configuring event receiving consists of the following steps:

1. Installing the pdAudit plugin.
2. Creating a KUMA collector for PostgreSQL events.

   To receive PostgreSQL events using rsyslog, in the collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] PostgreSQL pgAudit syslog** normalizer.
3. Installing a collector in the KUMA network infrastructure.
4. Configuring the event source server.
5. Verifying receipt of PostgreSQL events in the KUMA collector

   You can verify that the PostgreSQL event source server is correctly configured in the Searching for related events section of the KUMA Console.

Installing the pgAudit plugin

To install the pgAudit plugin:

1. On the OS command line, run the following commands as a user with administrator rights:

```
sudo apt update
sudo apt -y install postgresql-<PostgreSQL version>-pgaudit
```

You must select the plugin version to match the PostgreSQL version. For information about PostgreSQL versions and the matching plugin versions, see https://github.com/pgaudit/pgaudit#postgresql-version-compatibility.

```
Example:
sudo apt -y install postgresql-12-pgaudit
```

2. Find the postgres.conf configuration file. To do so, run the following command on the PostgreSQL command line:

   ```
   show data_directory
   ```

   The response will indicate the location of the configuration file.

3. Create a backup copy of the postgres.conf configuration file.

4. Open the postgres.conf file and copy or replace the values in it with the values listed below.

   ```
   ## pgAudit settings
   shared_preload_libraries = 'pgaudit'
   ## database logging settings
   log_destination = 'syslog'
   ## syslog facility
   syslog_facility = 'LOCAL0'
   ## event ident
   syslog_ident = 'Postgres'
   ## sequence numbers in syslog
   syslog_sequence_numbers = on
   ## split messages in syslog
   syslog_split_messages = off
   ## message encoding
   lc_messages = 'en_US.UTF-8'
   ## min message level for logging
   client_min_messages = log
   ## min error message level for logging
   log_min_error_statement = info
   ## log checkpoints (buffers, restarts)
   log_checkpoints = off
   ## log query duration
   log_duration = off
   ## error description level
   log_error_verbosity = default
   ## user connections logging
   log_connections = on
   ## user disconnections logging
   log_disconnections = on
   ## log prefix format
   ```

```
log_line_prefix = '%m|%a|%d|%p|%r|%i|%u| %e '
## log_statement
log_statement = 'none'
## hostname logging status. dns bane resolving affect
#performance!
log_hostname = off
## logging collector buffer status
#logging_collector = off
## pg audit settings
pgaudit.log_parameter = on
pgaudit.log='ROLE, DDL, MISC, FUNCTION'
```

5. Restart the PostgreSQL service using the command:

   ```
   sudo systemctl restart postgresql
   ```

6. To load the pgAudit plugin to PostgreSQL, run the following command on the PostgreSQL command line:

   ```
   CREATE EXTENSION pgaudit
   ```

The pgAudit plugin is installed.

Configuring a Syslog server to send events

The rsyslog service is used to transmit events from the server to KUMA.

To configure the sending of events from the server where PostgreSQL is installed to the collector:

1. To verify that the rsyslog service is installed on the event source server, run the following command as administrator:

   ```
   sudo systemctl status rsyslog.service
   ```

   If the rsyslog service is not installed on the server, install it by executing the following command:

   - for RPM-based distributions:

     ```
     yum install rsyslog
     ```

   - for Debian or Ubuntu-based distributions:

     ```
     apt-get install rsyslog
     ```

   Enable the service by executing the following command:

   ```
   sudo systemctl enable rsyslog.service
   sudo systemctl start rsyslog.service
   ```

2. In the /etc/rsyslog.d/ directory, create a pgsql-to-siem.conf file with the following content:

   ```
   If $programname contains 'Postgres' then @< IP address of the collector >:< port of the collector >
   For example:
   ```

```
If $programname contains 'Postgres' then @192.168.1.5:1514
```

If you want to send events via TCP, the contents of the file must be as follows:

```
If $programname contains 'Postgres' then @@192.168.1.5:2514
```

Save changes to the pgsql-to-siem.conf configuration file.

3. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/pgsql-to-siem.conf
   $RepeatedMsgReduction off
   ```

   Save changes to the /etc/rsyslog.conf configuration file.

4. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of IVK Kolchuga-K events

You can configure the receipt of events from the IVK Kolchuga-K system to the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the sending of IVK Kolchuga-K events to KUMA.

2. Creating a KUMA collector for receiving events from the IVK Kolchuga-K system.

   To receive IVK Kolchuga-K events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Kolchuga-K syslog** normalizer.

3. Installing a KUMA collector for receiving IVK Kolchuga-K events.

4. Verifying receipt of IVK Kolchuga-K events in KUMA.

   You can verify that the IVK Kolchuga-K event source is configured correctly in the Searching for related events section of the KUMA Console.

Configuring export of IVK Kolchuga-K events to KUMA

To configure the export of events of the IVK Kolchuga-K firewall via syslog to the KUMA collector:

1. Connect to the firewall over SSH with administrator rights.

2. Create a backup copy of the /etc/services and /etc/syslog.conf files.

3. In the /etc/syslog.conf configuration file, specify the FQDN or IP address of the KUMA collector. For example:

   . @kuma.example.com

   or

   . @192.168.0.100

   Save changes to the configuration file /etc/syslog.conf.

4. In the /etc/services configuration file, specify the port and protocol used by the KUMA collector. For example:

syslog 10514/udp

Save changes to the /etc/services configuration file.

5. Restart the syslog server of the firewall:

```
service syslogd restart
```

Configuring receipt of CryptoPro NGate events

You can configure the receipt of CryptoPro NGate events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of CryptoPro NGate events to KUMA.
2. Creating a KUMA collector for receiving CryptoPro NGate events.

   To receive CryptoPro NGate events using Syslog, in the collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] NGate syslog** normalizer.
3. Creating a KUMA collector for receiving CryptoPro NGate events.
4. Verifying receipt of CryptoPro NGate events in the KUMA collector.

   You can verify that the CryptoPro NGate event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of CryptoPro NGate events to KUMA

To configure the sending of events from CryptoPro NGate to KUMA:

1. Connect to the web interface of the NGate management system.
2. Connect remote syslog servers to the management system. To do so:

   a. Open the page with the list of syslog servers: **External Services** → **Syslog Server** → **Add Syslog Server**.

   b. Enter the settings of the syslog server and click ✔.
3. Assign syslog servers to the configuration for recording logs of the cluster. To do so:

   a. In the **Clusters** → **Summary** section, select the cluster that you want to configure.

   b. On the **Configurations** tab, click the **Configuration** control for the relevant cluster to go to the configuration settings page.

   c. In the **Syslog Servers** field of the configuration being configured, click **Assign**.

   d. Select the check boxes for syslog servers that you want to assign and click ✔.

   You can assign an unlimited number of servers.

To add new syslog servers, click +.

e. Publish the configuration to activate the new settings.

4. Assign syslog servers to the management system for recording Administrator activity logs. To do so:

   a. Select the **Management Center Settings** menu item and on the page that is displayed, under **Syslog servers**, click **Assign**.

   b. In the **Assign Syslog Servers to Management Center** window, select the check box for those syslog servers that you want to assign, then click ✓.

      You can assign an unlimited number of servers.

As a result, events of CryptoPro NGate are sent to KUMA.

Configuring receipt of Ideco UTM events

You can configure the receipt of Ideco UTM application events in KUMA via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring export of Ideco UTM events to KUMA.

2. Creating a KUMA collector for receiving Ideco UTM.

   To receive Ideco UTM events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Ideco UTM syslog** normalizer.

3. Creating a KUMA collector for receiving Ideco UTM events.

4. Verifying receipt of Ideco UTM events in KUMA.

   You can verify that the Ideco UTM event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of Ideco UTM events to KUMA

To configure the sending of events from Ideco UTM to KUMA:

1. Connect to the Ideco UTM web interface under a user account that has administrative privileges.

2. In the **System message forwarding** menu, move the **Syslog** toggle switch to the **enabled** position.

3. For the **IP address** setting, specify the IP address of the KUMA collector.

4. For the Port setting, enter the port that the KUMA collector is listening on.

5. Click **Save** to apply the changes.

The forwarding of Ideco UTM events to KUMA is configured.

Configuring receipt of KWTS events

You can configure the receipt of events from the Kaspersky Web Traffic Security (KWTS) web traffic analysis and filtering system in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring export of KWTS events to KUMA.
2. Creating a KUMA collector for receiving KWTS events.

   To receive KWTS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KWTS** normalizer.
3. Installing a KUMA collector for receiving KWTS events.
4. Verifying receipt of KWTS events in the KUMA collector.

   You can verify that KWTS event export is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of KWTS events to KUMA

To configure the export of KWTS events to KUMA:

1. Connect to the KWTS server over SSH as root.
2. Before making changes, create backup copies of the following files:
   - /opt/kaspersky/kwts/share/templates/core_settings/event_logger.json.template
   - /etc/rsyslog.conf
3. Make sure that the settings in the /opt/kaspersky/kwts/share/templates/core_settings/event_logger.json.template configuration file have the following values, and make changes if necessary:

   ```
   "siemSettings":
   {
   "enabled": true,
   "facility": "Local5",
   "logLevel": "Info",
   "formatting":
   {
   ```
4. Save your changes.
5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

   ```
   $WorkDirectory /var/lib/rsyslog
   $ActionQueueFileName ForwardToSIEM
   ```

```
$ActionQueueMaxDiskSpace 1g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1
local5.* @<< IP address of the KUMA collector >:< port of the collector >>
```

If you want to send events over TCP, the last line should be as follows:

```
local5.* @@<< IP address of the KUMA collector >:< port of the collector >>
```

6. Save your changes.

7. Restart the rsyslog service with the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

8. Go to the KWTS web interface, to the **Settings** – **Syslog** tab and enable the **Log information about traffic profile** option.

9. Click **Save**.

Configuring receipt of KLMS events

You can configure the receipt of events from the Kaspersky Linux Mail Server (KLMS) mail traffic analysis and filtering system to the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of KLMS events to KUMA

2. Creating a KUMA collector for receiving KLMS events

   To receive KLMS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KLMS syslog CEF** normalizer.

3. Installing a KUMA collector for receiving KLMS events.

4. Verifying receipt of KLMS events in the KUMA collector

   You can verify that the KLMS event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of KLMS events to KUMA

To configure the export of KLMS events to KUMA:

1. Connect to the KLMS server over SSH and go to the **Technical Support Mode** menu.

2. Use the klms-control utility to download the settings to the settings.xml file:

   ```
   sudo /opt/kaspersky/klms/bin/klms-control --get-settings EventLogger -n -f
   /tmp/settings.xml
   ```

3. Make sure that the settings in the /tmp/settings.xml file have the following values; make changes if necessary:

```
<siemSettings>
<enabled>1</enabled>
<facility>Local1</facility>
...
</siemSettings>
```

4. Apply settings with the following command:

```
sudo /opt/kaspersky/klms/bin/klms-control --set-settings EventLogger -n -f
/tmp/settings.xml
```

5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

```
$WorkDirectory /var/lib/rsyslog
$ActionQueueFileName ForwardToSIEM
$ActionQueueMaxDiskSpace 1g
$ActionQueueSaveOnShutdown on
$ActionQueueType LinkedList
$ActionResumeRetryCount -1
local1.* @<< IP address of the KUMA collector >:< port of the collector >>
```

If you want to send events over TCP, the last line should be as follows:

```
local1.* @@<< IP address of the KUMA collector >:< port of the collector >>
```

6. Save your changes.

7. Restart the rsyslog service with the following command:

```
sudo systemctl restart rsyslog.service
```

Configuring receipt of KSMG events

You can configure the receipt of events from the Kaspersky Secure Mail Gateway (KSMG) 1.1 mail traffic analysis and filtering system in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of KSMG events to KUMA

2. Creating a KUMA collector for receiving KSMG events

   To receive KSMG events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] KSMG** normalizer.

3. Installing a KUMA collector for receiving KSMG events.

4. Verifying receipt of KSMG events in the KUMA collector

   You can verify that the KSMG event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of KSMG events to KUMA

To configure the export of KSMG events to KUMA:

1. Connect to the KSMG server via SSH using an account with administrator rights.

2. Use the ksmg-control utility to download the settings to the settings.xml file:

   ```
   sudo /opt/kaspersky/ksmg/bin/ksmg-control --get-settings EventLogger -n -f
   /tmp/settings.xml
   ```

3. Make sure that the settings in the /tmp/settings.xml file have the following values; make changes if necessary:

   ```
   <siemSettings>
   <enabled>1</enabled>
   <facility>Local1</facility>
   ```

4. Apply settings with the following command:

   ```
   sudo /opt/kaspersky/ksmg/bin/ksmg-control --set-settings EventLogger -n -f
   /tmp/settings.xml
   ```

5. To send events via UDP, make the following changes to the /etc/rsyslog.conf configuration file:

   ```
   $WorkDirectory /var/lib/rsyslog
   $ActionQueueFileName ForwardToSIEM
   $ActionQueueMaxDiskSpace 1g
   $ActionQueueSaveOnShutdown on
   $ActionQueueType LinkedList
   $ActionResumeRetryCount -1
   local1.* @<< IP address of the KUMA collector >:< port of the collector >>
   ```

   If you want to send events over TCP, the last line should be as follows:

   ```
   local1.* @@<< IP address of the KUMA collector >:< port of the collector >>
   ```

6. Save your changes.

7. Restart the rsyslog service with the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of PT NAD events

You can configure the receipt of PT NAD events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring export of PT NAD events to KUMA.
2. Creating a KUMA collector for receiving PT NAD events.

   To receive PT NAD events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] PT NAD json** normalizer.
3. Installing a KUMA collector for receiving PT NAD events.
4. Verifying receipt of PT NAD events in the KUMA collector.

   You can verify that the PT NAD event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of PT NAD events to KUMA

Configuring the export of events from PT NAD 11 to KUMA over Syslog involves the following steps:

1. Configuring the ptdpi-worker@notifier module.
2. Configuring the sending of syslog messages with information about activities, attacks and indicators of compromise.

Configuring the ptdpi-worker@notifier module.

To enable the sending of information about detected information security threats, you must configure the ptdpi-worker@notifier module.

In a multi-server configuration, these instructions must be followed on the primary server.

To configure the ptdpi-worker@notifier module:

1. Open the /opt/ptsecurity/etc/ptdpi.settings.yaml file:

   ```
   sudo nano /opt/ptsecurity/etc/ptdpi.settings.yaml
   ```
2. In the **General settings** group of settings, uncomment the 'workers' setting and add 'notifier' to its list of values.

   For example:

   ```
   workers: ad alert dns es hosts notifier
   ```
3. To the end of the file, append a line of the form: notifier.yaml.nad_web_url: <URL of the PT NAD console>

   For example:

   ```
   notifier.yaml.nad_web_url: https://ptnad.example.com
   ```

   The ptdpi-worker@notifier module uses the specified URL to generate links to session and activity cards when sending messages.
4. Restart the sensor:

   ```
   sudo ptdpictl restart-all
   ```

The ptdpi-worker@notifier module is configured.

Configuring the sending of syslog messages with information about activities, attacks and indicators of compromise

The settings listed in the following instructions may not be present in the configuration file. If a setting is missing, you must add it to the file.

In a multi-server PT NAD configuration, edit the settings on the primary server.

To configure the sending of syslog messages with information about activities, attacks and indicators of compromise:

1. Open the /opt/ptsecurity/etc/ptdpi.settings.yaml file:

   ```
   sudo nano /opt/ptsecurity/etc/ptdpi.settings.yaml
   ```

2. By default, PT NAD sends activity information in Russian. To receive information in English, change the value of the notifier.yaml.syslog_notifier.locale setting to `en`.

   For example:

   ```
   notifier.yaml.syslog_notifier.locale: en
   ```

3. In the notifier.yaml.syslog_notifier.addresses setting, add a section with settings for sending events to KUMA.

   The <Connection name> setting can only contain Latin letters, numerals, and the underscore character.

   For the 'address' setting, specify the IP address of the KUMA collector.

   Other settings can be omitted, in which case the default values are used.

   ```
   notifier.yaml.syslog_notifier.addresses:
   <Connection name>:
   address: <For sending to a remote server, specify protocol: UDP (default) or TCP, address and port; for local connection, specify Unix domain socket>
   doc_types: [<Comma-separated message types ('alert' for information about attacks, 'detection' for activities, and 'reputation' for information about indicators of compromise). By default, all types of messages are sent>]
   facility: <Numeric value of the subject category>
   ident: <software tag>
   <Connection name>:
   ...
   ```

   The following is a sample configuration of sending syslog messages with information about activities, attacks, and indicators of compromise to two remote servers via TCP and UDP without writing to the local log:

   ```
   notifier.yaml.syslog_notifier.addresses:
   remote1:
   address: tcp://198.51.100.1:1514
   remote2:
   address: udp://198.51.100.2:2514
   ```

4. Save your changes in the /opt/ptsecurity/etc/ptdpi.settings.yaml.

5. Restart the ptdpi-worker@notifier module:

   ```
   sudo ptdpictl restart-worker notifier
   ```

The sending of events to KUMA via Syslog is configured.

Configuring receipt of events using the MariaDB Audit Plugin

KUMA allows auditing events using the MariaDB Audit Plugin. The plugin supports MySQL 5.7 and MariaDB. The audit plugin does not support MySQL 8. Detailed information about the plugin is available on the official MariaDB website.

We recommend using MariaDB Audit Plugin version 1.2 or later.

Configuring event receiving consists of the following steps:

1. Configuring the MariaDB Audit Plugin to send MySQL events and configuring the Syslog server to send events.

2. Configuring the MariaDB Audit Plugin to send MariaDB events and configuring the Syslog server to send events.

3. Creating a KUMA Collector for MySQL 5.7 and MariaDB Events.

   To receive MySQL 5.7 and MariaDB events using the MariaDB Audit Plugin, in the KUMA Collector Installation Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] MariaDB Audit Plugin syslog**.

4. Installing a collector in the KUMA network infrastructure.

5. Verifying receipt of MySQL and MariaDB events by the KUMA collector.

   To verify that the MySQL and MariaDB event source server is configured correctly, you can search for related events.

Configuring the MariaDB Audit Plugin to send MySQL events

The MariaDB Audit Plugin is supported for MySQL 5.7 versions up to 5.7.30 and is bundled with MariaDB.

To configure MySQL 5.7 event reporting using the MariaDB Audit Plugin:

1. Download the MariaDB distribution kit and extract it.

   You can download the MariaDB distribution kit from the official MariaDB website. The operating system of the MariaDB distribution must be the same as the operating system on which MySQL 5.7 is running.

2. Connect to MySQL 5.7 using an account with administrator rights by running the following command:

   ```
   mysql -u <username> -p
   ```

3. To get the directory where the MySQL 5.7 plugins are located, on the MySQL 5.7 command line, run the following command:

   ```
   SHOW GLOBAL VARIABLES LIKE 'plugin_dir'
   ```

4. In the directory obtained at step 3, copy the MariaDB Audit Plugin from `<directory to which the distribution kit was extracted>`/mariadb-server-`<version>`/lib/plugins/server_audit.so.

5. On the operating system command line, run the following command:

   ```
   chmod 755 <directory to which the distribution kit was extracted>server_audit.so
   ```

   For example:

```
chmod 755 /usr/lib64/mysql/plugin/server_audit.so
```

6. On the MySQL 5.7 command line, run the following command:

   ```
   install plugin server_audit soname 'server_audit.so'
   ```

7. Create a backup copy of the /etc/mysql/mysql.conf.d/mysqld.cnf configuration file.

8. In the configuration file /etc/mysql/mysql.conf.d/mysqld.cnf, in the `[mysqld]` section, add the following lines:

   ```
   server_audit_logging=1
   server_audit_events=connect,table,query_ddl,query_dml,query_dcl
   server_audit_output_type=SYSLOG
   server_audit_syslog_facility=LOG_SYSLOG
   ```

   If you want to disable event export for certain audit event groups, remove some of the values from the `server_audit_events` setting. Descriptions of settings are available on the MariaDB Audit Plugin vendor's website.

9. Save changes to the configuration file.

10. Restart the MariaDB service by running one of the following commands:

    - `systemctl restart mysqld` for a system with systemd initialization.
    - `service mysqld restart` for a system with init initialization.

MariaDB Audit Plugin for MySQL 5.7 is configured. If necessary, you can run the following commands on the MySQL 5.7 command line:

- `show plugins` to check the list of current plugins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to check the current audit settings.

Configuring the MariaDB Audit Plugin to send MariaDB Events

The MariaDB Audit Plugin is included in the MariaDB distribution kit starting with versions 5.5.37 and 10.0.10.

To configure MariaDB event export using the MariaDB Audit Plugin:

1. Connect to MariaDB using an account with administrator rights by running the following command:

   ```
   mysql -u <username> -p
   ```

2. To check if the plugin is present in the directory where operating system plugins are located, run the following command on the MariaDB command line:

   ```
   SHOW GLOBAL VARIABLES LIKE 'plugin_dir'
   ```

3. On the operating system command line, run the following command:

   ```
   ll <directory obtained by the previous command> | grep server_audit.so
   ```

   If the command output is empty and the plugin is not present in the directory, you can either copy the MariaDB Audit Plugin to that directory or use a newer version of MariaDB.

4. On the MariaDB command line, run the following command:

   ```
   install plugin server_audit soname 'server_audit.so'
   ```

5. Create a backup copy of the /etc/mysql/my.cnf configuration file.

6. In the /etc/mysql/my.cnf configuration file, in the `[mysqld]` section, add the following lines:

   ```
   server_audit_logging=1
   server_audit_events=connect,table,query_ddl,query_dml,query_dcl
   server_audit_output_type=SYSLOG
   server_audit_syslog_facility=LOG_SYSLOG
   ```

   If you want to disable event export for certain audit event groups, remove some of the values from the `server_audit_events` setting. Descriptions of settings are available on the MariaDB Audit Plugin vendor's website.

7. Save changes to the configuration file.

8. Restart the MariaDB service by running one of the following commands:

   - `systemctl restart mariadb` for a system with systemd initialization.
   - `service mariadb restart` for a system with init initialization.

   MariaDB Audit Plugin for MariaDB is configured. If necessary, you can run the following commands on the MariaDB command line:

- `show plugins` to check the list of current plugins.
- `SHOW GLOBAL VARIABLES LIKE 'server_audit%'` to check the current audit settings.

Configuring a Syslog server to send events

The rsyslog service is used to transmit events from the server to the collector.

To configure the sending of events from the server where MySQL or MariaDB is installed to the collector:

1. Before making any changes, create a backup copy of the /etc/rsyslog.conf configuration file.

2. To send events via UDP, add the following line to the /etc/rsyslog.conf configuration file:

   ```
   *.* @<IP address of the KUMA collector>:<port of the KUMA collector>
   ```

   For example:

   ```
   *.* @192.168.1.5:1514
   ```

   If you want to send events over TCP, the line should be as follows:

   ```
   *.* @@192.168.1.5:2514
   ```

   Save changes to the /etc/rsyslog.conf configuration file.

3. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of Apache Cassandra events

KUMA allows receiving information about Apache Cassandra events.

Configuring event receiving consists of the following steps:

1. Configuring Apache Cassandra event logging in KUMA.

2. Creating a KUMA collector for Apache Cassandra events.

   To receive Apache Cassandra events, in the KUMA Collector Installation Wizard, at the **Transport** step, select a **file** type connector; at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Apache Cassandra file**.

3. Installing a collector in the KUMA network infrastructure.

4. Verifying receipt of Apache Cassandra events in the KUMA collector.

   To verify that the Apache Cassandra event source server is configured correctly, you can search for related events.

Configuring Apache Cassandra event logging in KUMA

To configuring Apache Cassandra event logging in KUMA:

1. Make sure that the server where Apache Cassandra is installed has 5 GB of free disk space.

2. Connect to the Apache Cassandra server using an account with administrator rights.

3. Before making changes, create backup copies of the following configuration files:

   - /etc/cassandra/cassandra.yaml

   - /etc/cassandra/logback.xml

4. Make sure that the settings in the /etc/cassandra/cassandra.yaml configuration file have the following values; make changes if necessary:

   a. in the `audit_logging_options` section, set the `enabled` setting to `true`.

   b. in the `logger` section, set the `class_name` setting to `FileAuditLogger`.

5. Add the following lines to the /etc/cassandra/logback.xml configuration file:

```
<!-- Audit Logging (FileAuditLogger) rolling file appender to audit.log -->
<appender name="AUDIT" class="ch.qos.logback.core.rolling.RollingFileAppender">
<file>${cassandra.logdir}/audit/audit.log</file>
<rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
<!-- rollover daily -->
<fileNamePattern>${cassandra.logdir}/audit/audit.log.%d{yyyy-MM-
dd}.%i.zip</fileNamePattern>
```

```
<!-- each file should be at most 50MB, keep 30 days worth of history, but at most 5GB
-->
<maxFileSize>50MB</maxFileSize>
<maxHistory>30</maxHistory>
<totalSizeCap>5GB</totalSizeCap>
</rollingPolicy>
<encoder>
<pattern>%-5level [%thread] %date{ISO8601} %F:%L - %replace(%msg){'\n', '
'}%n</pattern>
</encoder>
</appender>
<!-- Audit Logging additivity to redirect audt logging events to audit/audit.log -->
<logger name="org.apache.cassandra.audit" additivity="false" level="INFO">
<appender-ref ref="AUDIT"/>
</logger>
```

6. Save changes to the configuration file.

7. Restart the Apache Cassandra service using the following commands:

   a. `sudo systemctl stop cassandra.service`

   b. `sudo systemctl start cassandra.service`

8. After restarting, check the status of Apache Cassandra using the following command:

   ```
   sudo systemctl status cassandra.service
   ```

   Make sure that the command output contains the following sequence of characters:

   ```
   Active: active (running)
   ```

Apache Cassandra event export is configured. Events are located in the /var/log/cassandra/audit/ directory, in the audit.log file (${cassandra.logdir}/audit/audit.log).

Configuring receipt of FreeIPA events

You can configure the receipt of FreeIPA events in KUMA via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring export of FreeIPA events to KUMA.

2. Creating a KUMA collector for receiving FreeIPA events.

   To receive FreeIPA events, in the KUMA Collector Setup Wizard, at the **Event parsing** step, in the **Normalizer** field, select **[OOTB] FreeIPA**.

3. Installing the KUMA collector in the network infrastructure.

4. Verifying receipt of FreeIPA events by KUMA.

   To verify that the FreeIPA event source server is configured correctly, you can search for related events.

Configuring export of FreeIPA events to KUMA

To configure the export of FreeIPA events to KUMA via the Syslog protocol in JSON format:

1. Connect to the FreeIPA server via SSH using an account with administrator rights.
2. In the /etc/rsyslog.d/ directory, create a file named freeipa-to-siem.conf.
3. Add the following lines to the /etc/rsyslog.d/freeipa-to-siem.conf configuration file:

```
$ModLoad imfile
$InputFileName /var/log/httpd/error_log
$InputFileTag tag_FreeIPA_log:
$InputFileStateFile FreeIPA_log
$InputRunFileMonitor
$InputFileName /var/log/dirsrv/slapd-*/audit
$InputFileTag tag_FreeIPA_log:
$InputFileStateFile FreeIPA_log_audit
$InputRunFileMonitor
$InputFileName /var/log/dirsrv/slapd-*/errors
$InputFileTag tag_FreeIPA_log:
$InputFileStateFile FreeIPA_log_errors
$InputRunFileMonitor
$InputFileName /var/log/dirsrv/slapd-*/access
$InputFileTag tag_FreeIPA_log:
$InputFileStateFile FreeIPA_log_access
$InputRunFileMonitor
$InputFileName /var/log/krb5kdc.log
$InputFileTag tag_FreeIPA_log:
$InputFileStateFile FreeIPA_log_krb5kdc
$InputRunFileMonitor
template(name="ls_json" type="list" option.json="on") {
constant(value="{")
constant(value="\"@timestamp\":\"") property(name="timegenerated"
dateFormat="rfc3339")
constant(value="\",\"@version\":\"1")
constant(value="\",\"message\":\"") property(name="msg")
constant(value="\",\"host\":\"") property(name="fromhost")
constant(value="\",\"host_ip\":\"") property(name="fromhost-ip")
constant(value="\",\"logsource\":\"") property(name="fromhost")
constant(value="\",\"severity_label\":\"") property(name="syslogseverity-text")
constant(value="\",\"severity\":\"") property(name="syslogseverity")
```

```
constant(value="\",\"facility_label\":\"") property(name="syslogfacility-text")
constant(value="\",\"facility\":\"") property(name="syslogfacility")
constant(value="\",\"program\":\"") property(name="programname")
constant(value="\",\"pid\":\"") property(name="procid")
constant(value="\",\"syslogtag\":\"") property(name="syslogtag")
constant(value="\"}\n")
}
if $syslogtag contains 'tag_FreeIPA_log' then {
*.* action(type="omfwd" target="< IP address of the KUMA collector >" port="< port of
the KUMA collector >" protocol="< udp or tcp >" template="ls_json")
stop
}
```

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/freeipa-to-siem.conf
   $RepeatedMsgReduction off
   ```

5. Save changes to the configuration file.

6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Configuring receipt of VipNet TIAS events

You can configure the receipt of ViPNet TIAS events in KUMA via the Syslog protocol.

Configuring event receiving consists of the following steps:

1. Configuring export of ViPNet TIAS events to KUMA.

2. Creating a KUMA collector for receiving ViPNet TIAS events.

   To receive ViPNet TIAS events using Syslog, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Syslog-CEF** normalizer.

3. Installing a KUMA collector for receiving ViPNet TIAS events.

4. Verifying receipt of ViPNet TIAS events in KUMA.

   You can verify that ViPNet TIAS event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of ViPNet TIAS events to KUMA

To configure the export of ViPNet TIAS events to KUMA via the syslog protocol:

1. Connect to the ViPNet TIAS web interface under a user account with administrator rights.
2. Go to the **Management** – **Integrations** section.
3. On the **Integration** page, go to the Syslog tab.
4. In the toolbar of the list of receiving servers, click **New server**.
5. This opens the new server card; in that card:
   1. In the **Server address** field, enter the IP address or domain name of the KUMA collector.

      For example, 10.1.2.3 or syslog.siem.ru
   2. In the **Port** field, specify the inbound port of the KUMA collector. The default port number is 514.
   3. In the **Protocol** list, select the transport layer protocol that the KUMA collector is listening on. UDP is selected by default.
   4. In the **Organization** list, use the check boxes to select the organizations of the ViPNet TIAS infrastructure.

      Messages are sent only for incidents detected based on events received from sensors of selected organizations of the infrastructure.
   5. In the **Status** list, use check boxes to select incident statuses.

      Messages are sent only when selected statuses are assigned to incidents.
   6. In the **Severity level** list, use check boxes to select the severity levels of the incidents.

      Messages are sent only about incidents with the selected severity levels. By default, only the high severity level is selected in the list.
   7. In the **UI language** list, select the language in which you want to receive information about incidents in messages. Russian is selected by default.
6. Click **Add**.
7. In the toolbar of the list, set the **Do not send incident information in CEF format** toggle switch to enabled.

   As a result, when new incidents are detected or the statuses of previously detected incidents change, depending on the statuses selected during configuration, the corresponding information is sent to the specified addresses of receiving servers via the syslog protocol in CEF format.
8. Click **Save changes**.

Export of events to the KUMA collector is configured.

Configuring receipt of Nextcloud events

You can configure the receipt of Nextcloud 26.0.4 events in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring audit of Nextcloud events.
2. Configuring a Syslog server to send events.

   The rsyslog service is used to transmit events from the server to the collector.
3. Creating a KUMA collector for receiving Nextcloud events.

   To receive Nextcloud events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Nextcloud syslog** normalizer, and at the **Transport** step select the **tcp** or **udp** connector type.
4. Installing KUMA collector for receiving Nextcloud events
5. Verifying receipt of Nextcloud events in the KUMA collector

   You can verify that the Nextcloud event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring audit of Nextcloud events

To configure the export of Nextcloud events to KUMA:

1. On the server where Nextcloud is installed, create a backup copy of the /home/localuser/www/nextcloud/config/config.php configuration file.
2. Edit the /home/localuser/www/nextcloud/config/config.php Nextcloud configuration file.
3. Edit the settings as follows:

   ```
   'log_type' => 'syslog',
   'syslog_tag' => 'Nextcloud',
   'logfile' => '',
   'loglevel' => 0,
   'log.condition' => [
   'apps' => ['admin_audit'],
   ],
   ```
4. Restart the Nextcloud service:

   ```
   sudo service restart nextcloud
   ```

Export of events to the KUMA collector is configured.

Configuring a Syslog server to send Nextcloud events

To configure the sending of events from the server where Nextcloud is installed to the collector:

1. In the /etc/rsyslog.d/ directory, create a Nextcloud-to-siem.conf file with the following content:

   ```
   If $programname contains 'Nextcloud' then @<IP address of the collector>:<port of the collector>
   ```

   Example:

   ```
   If $programname contains 'Nextcloud' then @192.168.1.5:1514
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'Nextcloud' then @<IP address of the collector>:<port of the collector>
   ```

2. Save changes to the Nextcloud-to-siem.conf configuration file.

3. Create a backup copy of the /etc/rsyslog.conf file.

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/Nextcloud-to-siem.conf
   ```

   ```
   $RepeatedMsgReduction off
   ```

5. Save your changes.

6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

The export of Nextcloud events to the collector is configured.

Configuring receipt of Snort events

You can configure the receipt of Snort 3 events in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring logging of Snort events.

2. Creating a KUMA collector for receiving Snort events.

   To receive Snort events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Snort 3 json file** normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing a KUMA collector for receiving Snort events

4. Verifying receipt of Snort events in the KUMA collector

   You can verify that the Snort event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring logging of Snort events

Make sure that the server running Snort has at least 500 MB of free disk space for storing a single Snort event log.
When the log reaches 500 MB, Snort automatically creates a new file with a name that includes the current time in unixtime format.
We recommend monitoring disk space usage.

To configure Snort event logging:

1. Connect to the server where Snort is installed using an account with administrative privileges.

2. Edit the Snort configuration file. To do so, run the following command on the command line:

   ```
   sudo vi /usr/local/etc/snort/snort.lua
   ```

3. In the configuration file, edit the alert_json block:

   ```
   alert_json =
   {
   file = true,
   limit = 500,
   fields = 'seconds action class b64_data dir dst_addr dst_ap dst_port eth_dst eth_len \
   eth_src eth_type gid icmp_code icmp_id icmp_seq icmp_type iface ip_id ip_len msg mpls
   \
   pkt_gen pkt_len pkt_num priority proto rev rule service sid src_addr src_ap src_port \
   target tcp_ack tcp_flags tcp_len tcp_seq tcp_win tos ttl udp_len vlan timestamp',
   }
   ```

4. To complete the configuration, run the following command:

   ```
   sudo /usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -s 65535 -k none -l
   /var/log/snort -i <name of the interface that Snort is listening on> -m 0x1b
   ```

   As a result, Snort events are logged to /var/log/snort/alert_json.txt.

Configuring receipt of Suricata events

You can configure the receipt of Suricata 7.0.1 events in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring export of Suricata events to KUMA

2. Creating a KUMA collector for receiving Suricata events.

   To receive Suricata events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Suricata json file** normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing KUMA collector for receiving Suricata events

4. Verifying receipt of Suricata events in the KUMA collector

   You can verify that the Suricata event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring logging of Suricata events.

To configure Suricata event logging:

1. Connect via SSH to the server that has administrative user accounts.

2. Create a backup copy of the /etc/suricata/suricata.yaml file.

3. Set the following values in the eve-log section of the /etc/suricata/suricata.yaml configuration file:

```
- eve-log:
enabled: yes
filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
filename: eve.json
```

4. Save your changes to the /etc/suricata/suricata.yaml configuration file.

As a result, Suricata events are logged to the /usr/local/var/log/suricata/eve.json file.

Suricata does not support limiting the size of the eve.json event file. If necessary, you can manage the log size by using rotation. For example, to configure hourly log rotation, add the following lines to the configuration file:

```
outputs:

- eve-log:

filename: eve-%Y-%m-%d-%H:%M.json

rotate-interval: hour
```

Configuring receipt of FreeRADIUS events

You can configure the receipt of FreeRADIUS 3.0.26 events in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring audit of FreeRADIUS events.

2. Configuring a Syslog server to send FreeRADIUS events.

3. Creating a KUMA collector for receiving FreeRADIUS events.

   To receive FreeRADIUS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] FreeRADIUS syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.

4. Installing KUMA collector for receiving FreeRADIUS events.

5. Verifying receipt of FreeRADIUS events in the KUMA collector.

   You can verify that the FreeRADIUS event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring audit of FreeRADIUS events

To configure event audit in the FreeRADIUS system:

1. Connect to the server where the FreeRADIUS system is installed with a user account with administrative privileges.

2. Create a backup copy of the FreeRADIUS configuration file:

   ```
   sudo cp /etc/freeradius/3.0/radiusd.conf /etc/freeradius /3.0/radiusd.conf.bak
   ```

3. Open the FreeRADIUS configuration file for editing:

   ```
   sudo nano /etc/freeradius/3.0/radiusd.conf
   ```

4. In the 'log' section, edit the settings as follows:

   ```
   destination = syslog
   syslog_facility = daemon
   stripped_names = no
   auth = yes
   auth_badpass = yes
   auth_goodpass = yes
   ```

5. Save the configuration file.

FreeRADIUS event audit is configured.

Configuring a Syslog server to send FreeRADIUS events

The rsyslog service is used to transmit events from the FreeRADIUS server to the KUMA collector.

To configure the sending of events from the server where FreeRADIUS is installed to the collector:

1. In the /etc/rsyslog.d/ directory, create the FreeRADIUS-to-siem.conf file and add the following line to it:

   ```
   If $programname contains 'radiusd' then @<IP address of the collector>:<port of the collector>
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

   ```
   If $programname contains 'radiusd' then @<IP address of the collector>:<port of the collector>
   ```

2. Create a backup copy of the /etc/rsyslog.conf file.

3. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/FreeRADIUS-to-siem.conf
   ```

```
$RepeatedMsgReduction off
```

4. Save your changes.

5. Restart the rsyslog service:

   ```
   sudo systemctl restart rsyslog.service
   ```

The export of events from the FreeRADIUS server to the KUMA collector is configured.

Configuring receipt of VMware vCenter events

You can configure the receipt of VMware vCenter events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring the connection to VMware vCenter.

2. Creating a KUMA collector for receiving VMware vCenter events.

   To receive VMware vCenter events, in the collector Installation Wizard, at the **Transport** step, select the vmware connector type. Specify the required settings:

   - The URL at which the VMware API is available, for example, https://vmware-server.com:6440.

   - VMware credentials — a secret that specifies the username and password for connecting to the VMware API.

   At the **Event parsing** step, select the **[OOTB] VMware vCenter API** normalizer.

3. Installing a KUMA collector for receiving VMware vCenter events.

4. Verifying receipt of VMware vCenter events in the KUMA collector.

   You can verify that the VMware vCenter event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring the connection to VMware vCenter

To configure a connection to VMware vCenter to receive events:

1. Connect to the VMware vCenter web interface under a user account that has administrative privileges.

2. Go to the **Security&Users** section and select **Users**.

3. Create a user account.

4. Go to the **Roles** section and assign the **Read-only: See details of objects role, but not make changes** role to the created account.

   You will use the credentials of this user account in the secret of the collector.

   For details about creating user accounts, refer to the VMware vCenter documentation.

The connection to VMware vCenter for receiving events is configured.

Configuring receipt of zVirt events

You can configure the receipt of zVirt 3.1 events in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring export of zVirt events to KUMA.
2. Creating a KUMA collector for receiving zVirt events.

   To receive zVirt events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] OrionSoft zVirt syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.
3. Installing KUMA collector for receiving zVirt events
4. Verifying receipt of zVirt events in the KUMA collector

   You can verify that the zVirt event source server is correctly configured in the Searching for related events section of the KUMA Console.

Configuring export of zVirt events

ZVirt can send events to external systems in Hosted Engine installation mode.

To configure the export of zVirt events to KUMA:

1. In the zVirt web interface, under **Resources**, select **Virtual machines**.
2. Select the machine that is running the HostedEngine virtual machine and click **Edit**.
3. In the **Edit virtual machine** window, go to the **Logging** section.
4. Select the **Determine Syslog server address** check box.
5. In the text box, enter the collector information in the following format: `<IP address or FQDN of the KUMA collector>: <port of the KUMA collector>`.
6. If you want to use TCP instead of UDP for sending logs, select the **Use TCP connection** check box.

Event export is configured.

Configuring receipt of Zeek IDS events

You can configure the receipt of Zeek IDS 1.8 events in KUMA.

Configuring event receiving consists of the following steps:

1. Conversion of the Zeek IDS event log format.

   The KUMA normalizer supports Zeek IDS logs in the JSON format. To send events to the KUMA normalizer, log files must be converted to the JSON format.

2. Creating a KUMA collector for receiving Zeek IDS events.

   To receive Zeek IDS events, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] ZEEK IDS json** file normalizer, and at the **Transport** step, select the **file** connector type.

3. Installing KUMA collector for receiving Zeek IDS events

4. Verifying receipt of Zeek IDS events in the KUMA collector

   You can verify that the Zeek IDS event source server is correctly configured in the Searching for related events section of the KUMA Console.

Conversion of the Zeek IDS event log format

By default, Zeek IDS events are logged in files in the /opt/zeek/logs/current directory.

The [OOTB] ZEEK IDS json file normalizer supports Zeek IDS logs in the JSON format. To send events to the KUMA normalizer, log files must be converted to the JSON format.

This procedure must be repeated every time before receiving Zeek IDS events.

To convert the Zeek IDS event log format:

1. Connect to the server where Zeek IDS is installed with a user account with administrative privileges.

2. Create the directory where JSON event logs must be stored:

   ```
   sudo mkdir /opt/zeek/logs/zeek-json
   ```

3. Change to this directory:

   ```
   sudo cd /opt/zeek/logs/zeek-json
   ```

4. Run the command that uses the jq utility to convert the original event log format to the target format:

   ```
   jq . -c <path to the log file to be converted to a different format> >> <new file name> .log
   ```

   Example:
   ```
   jq . -c /opt/zeek/logs/current/conn.log >> conn.log
   ```

   As a result of running the command, a new file is created in the /opt/zeek/logs/zeek-json directory if this file did not exist before. If the file was already present in the current directory, new information is appended to the end of the file.

Configuring DNS server event reception using the ETW connector

The Event Tracing for Windows connector (hereinafter also referred to as the ETW connector) is a mechanism for logging events generated by applications and drivers on the DNS server. You can use the ETW connector to troubleshoot errors during development or to look for malicious activity.

The impact of the ETW connector on DNS server performance is insignificant. For example, a DNS server running on modern hardware and getting up to 100,000 queries per second (QPS) may experience a 5% performance drop while using the ETW connector. If the DNS server gets up to 50,000 requests per second, no performance drop is observed. We recommend monitoring DNS server performance when using the ETW connector, regardless of the number of requests per second.

> By default, you can use the ETW connector on Windows Server 2016 or later. The ETW connector is also supported by Windows Server 2012 R2 if the update for event logging and change auditing is installed. The update is available on the Microsoft Support website.

The ETW connector consists of the following components:

- *Provider* is the software that generates events and sends the events to the ETW connector. For example, Windows kernels or device drivers can act as providers. When working with code, developers must specify which events the providers must send to the ETW connector. An event may represent the execution of a function that the developer considers important, for example, a function that allows access to the Security Account Manager (SAM).

- *Consumer* is the software that receives and uses events generated by providers from the ETW connector. For example, KUMA can act as a consumer.

- *Controller* is the software that manages the interaction between providers and consumers. For example, the Logman or Wevtutil utilities can be controllers. Providers register with the controller to send events to consumers. The controller can enable or disable a provider. If a provider is disabled, it does not generate events.

  Controllers use trace sessions for communication between providers and consumers. Trace sessions are also used for filtering data based on specified parameters because consumers may need different events.

Configuring DNS server event reception using the ETW connector involves the following steps:

1. Configuration on the Windows side.

2. Creating a KUMA collector.

   When creating a KUMA collector, follow these steps:

   a. At step 2 of the Collector Installation Wizard:

      1. In the **Type** drop-down list, select the **tcp** connector type. You can also specify the **http** connector type and other connector types with verification for secure transmission.

      2. In the **URL** field, enter the FQDN and port number on which the KUMA collector will listen for a connection from the KUMA agent. You can specify any unoccupied port number.

      3. In the **Delimiter** field, enter \n.

b. At the step 3 of the Collector Installation Wizard, in the **Normalizer** drop-down list, select a normalizer. We recommend selecting the predefined extended normalizer for Windows events, **[OOTB] Microsoft DNS ETW logs json**.

c. At step 7 of the Collector Installation Wizard, add a **Storage** type destination for storing events. If you plan to use event correlation, you also need to add a **Correlator** type destination.

d. At step 8 of the Collector Installation Wizard, click **Create and save service**, and in the lower part of the window, copy the command for installing the KUMA collector on the server.

3. Installing the KUMA collector on the server.

   Do the following:

   a. Connect to the KUMA command line interface using a user account with root privileges.

   b. Install the KUMA collector by running the command that you copied at step 8 of the Collector Installation Wizard.

   c. If you want to add the KUMA collector port to the firewall exclusions and update the firewall settings, run the following commands:

      1. `firewall-cmd --add-port=<collector port number>/tcp --permanent`

      2. `firewall-cmd --reload`

   The KUMA collector is installed and the status of the KUMA collector service is changed to green in the KUMA Console.

4. Creating a KUMA agent.

   When creating a KUMA agent, follow these steps:

   a. Go to the **Connection 1** tab.

   b. Under **Connector**, in the **Connector** drop-down list, select **Create** and specify the following settings:

      1. In the **Type** drop-down list, select the **etw** connector type.

      2. In the **Session name** field, enter the provider name that you specified when you configured the reception of DNS server events using the ETW connector on the Windows side.

   c. Under **Destinations**, in the **Destination** drop-down list, select **Create** and specify the following settings:

      1. In the **Type** drop-down list, select the tcp destination type.

      2. In the **URL** field, enter the FQDN and port number on which the KUMA collector will listen for a connection from the KUMA agent. The value must match the value that you specified at step 2 of the Collector Installation Wizard.

   d. Go to the **Advanced settings** tab, and in the **Disk buffer size limit** field, enter 1073741824.

5. Creating a KUMA agent service.

   You need to copy the ID of the created KUMA agent service. To do that, click the KUMA agent service, and then select **Copy ID** in the context menu.

6. Creating an account for the KUMA agent.

   Create a domain or local Windows user account for running the KUMA agent and reading the analytic log. You need to add the created user account to the **Performance Log Users** group and grant the **Log on service** permission to that user account.

7. Installing a KUMA agent on a Windows server.

   You need to install the KUMA agent on the Windows server that will be receiving events from the provider. To do so:

   a. Add the FQDN of the KUMA Core server to the hosts file on the Windows server or to the DNS server.

   b. Create the C:\Users\<user name>\Desktop\KUMA folder on the Windows server.

   c. Copy the kuma.exe file from the KUMA installation package archive to the C:\Users\<user name>\Desktop\KUMA folder.

   d. Run the command interpreter as administrator.

   e. Change to the C:\Users\<user name>\Desktop\KUMA folder and run the following command:

      ```
      C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --core https://<DOMAIN-NAME-KUMA-CORE-Server>:7210 --id <KUMA agent service ID>
      ```

      In the KUMA Console, in the **Resources** → **Active services** section, make sure that the KUMA agent service is running and its status is now green, and then abort the command.

   f. Start the KUMA Agent installation in one of the following ways:

      - If you want to start the KUMA agent installation using a domain user account, run the following command:

        ```
        C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --core https://<DOMAIN-NAME-KUMA-CORE-Server>:7210 --id <KUMA agent service ID> –-user <domain>\<user account name for the KUMA agent> --install
        ```

      - If you want to start the KUMA agent installation using a local user account, run the following command:

        ```
        C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --core https://<DOMAIN-NAME-KUMA-CORE-Server>:7210 --id <KUMA agent service ID> –-user <user account name for the KUMA agent> --install
        ```

      You will need to enter the password of the KUMA agent user account.

   The KUMA Windows Agent service <KUMA agent service ID> is installed on the Windows server. In the KUMA Console, in the **Resources** → **Active services** section, if the KUMA agent service is not running and has the red status, you need to make sure that port 7210 is available, as well as the Windows collector port in the direction from the KUMA agent to the KUMA collector.

   To remove the KUMA agent service on the Windows server, run the following command:

   ```
   C:\Users\<user name>\Desktop\KUMA>kuma.exe agent --id <KUMA agent service ID> --uninstall
   ```

8. Verifying receipt of DNS server events in the KUMA collector.

   You can verify that you have correctly configured the reception of DNS server events using the ETW connector in the **Searching for related events** section of the KUMA Console.

Configuration on the Windows side

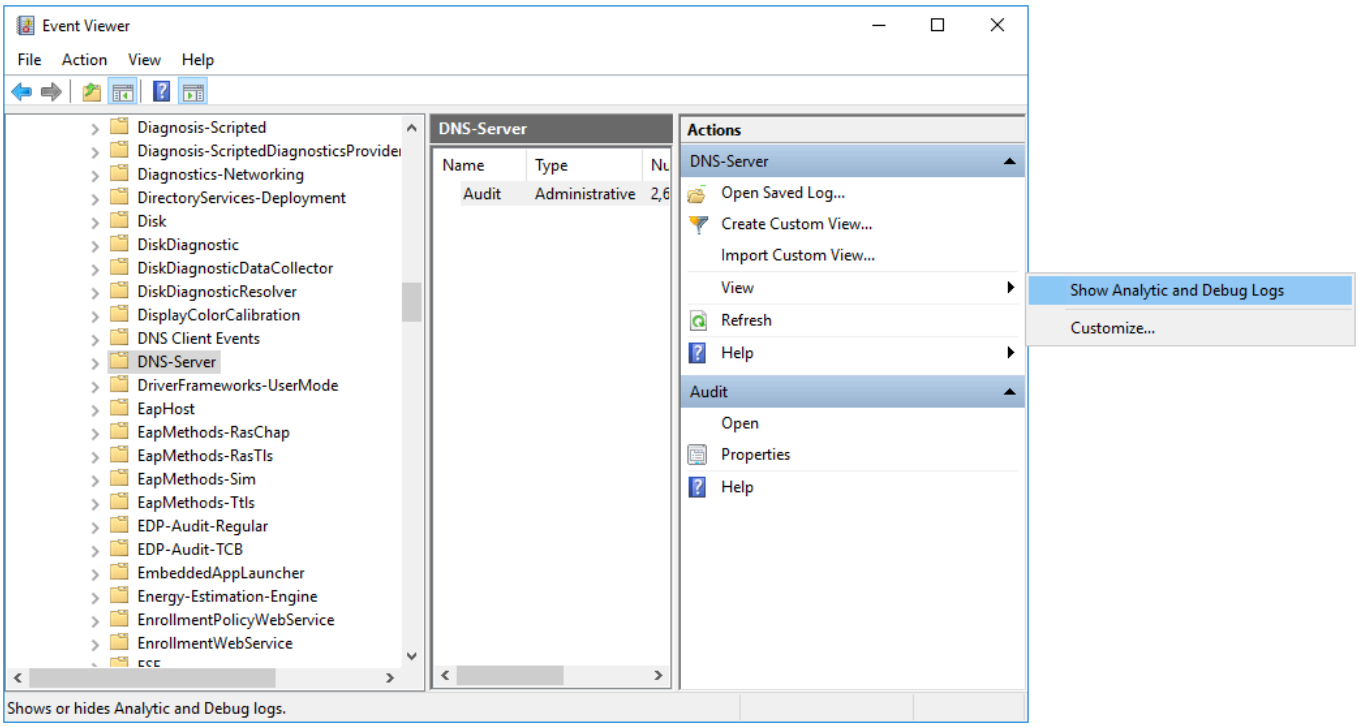
To configure the reception of DNS server events using the ETW connector on the Windows side:

1. Start the Event viewer by running the following command:

   eventvwr.msc

2. In the window that opens, go to the **Applications and Services Logs** → **Microsoft** → **Windows** → **DNS-Server** folder.

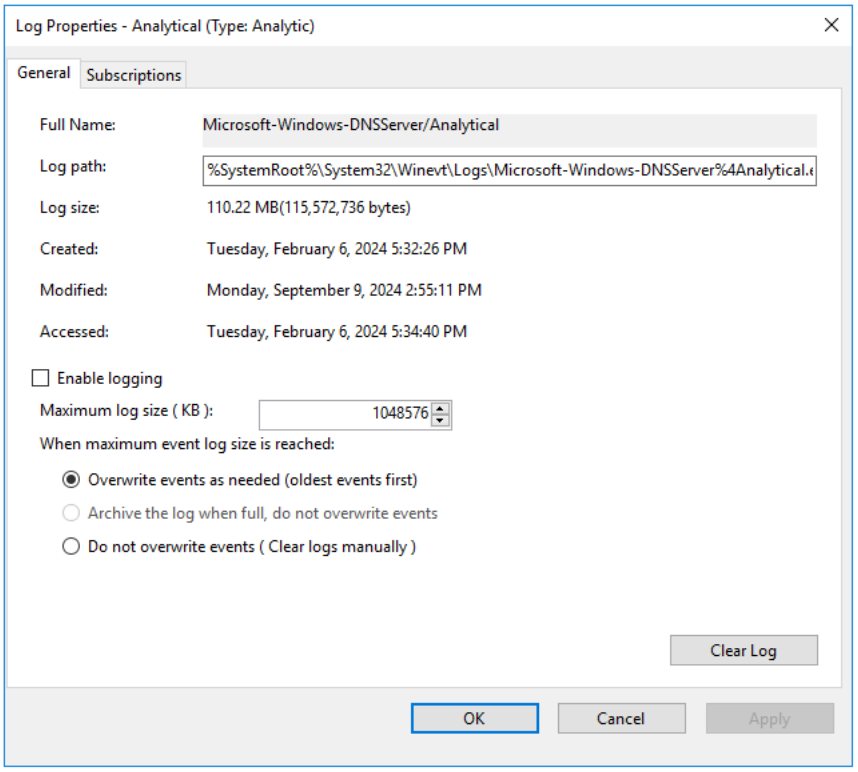
3. Open the context menu of the **DNS-Server** folder and select **View** → **Show Analytic and Debug Logs**.



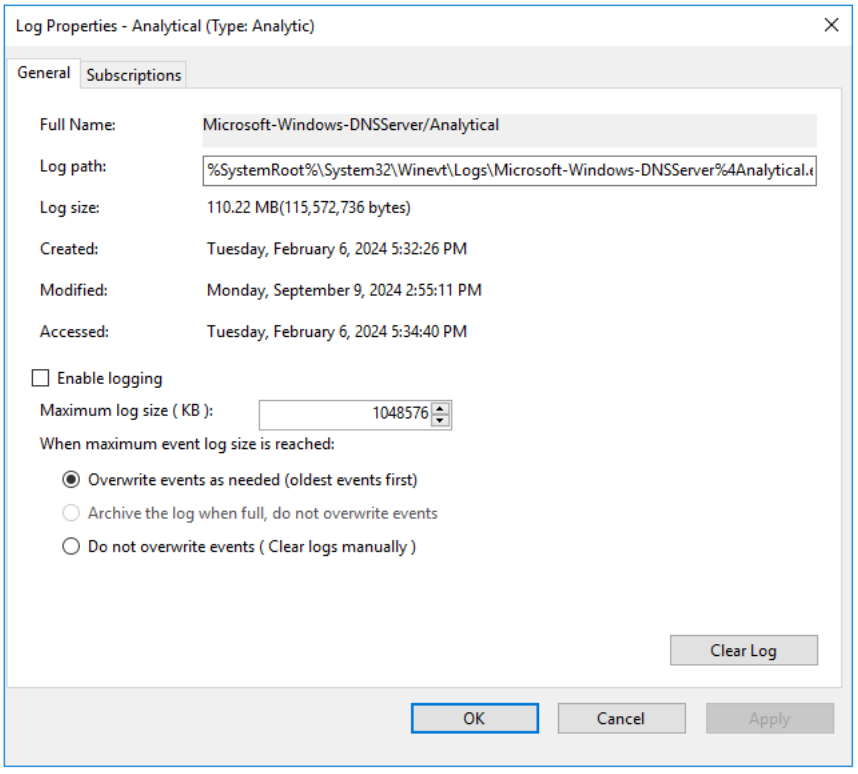
   The **Audit** debug log and **Analytical** log are displayed.

4. Configure the analytic log:

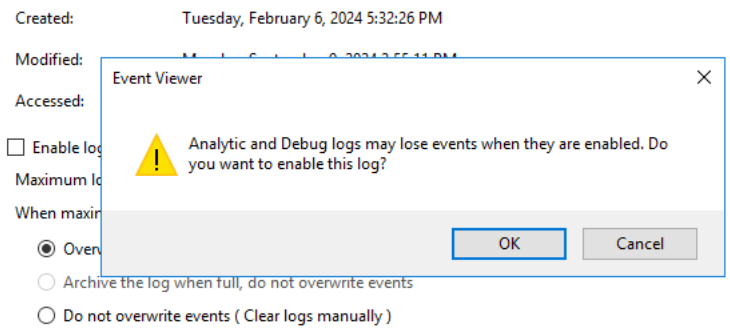
a. Open the context menu of the **Analytical** log and select **Properties**.



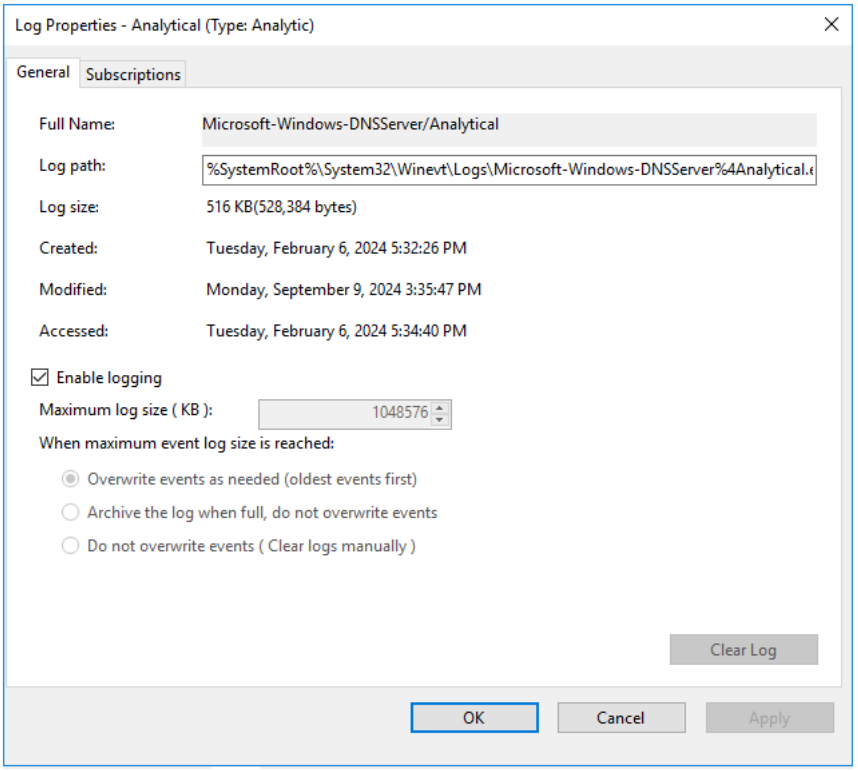
a. In the window that opens, make sure that in the **Max Log Size (KB)** field, the value is 1048576.



a. Select the **Enable logging** check box and in the confirmation window, click **OK**.

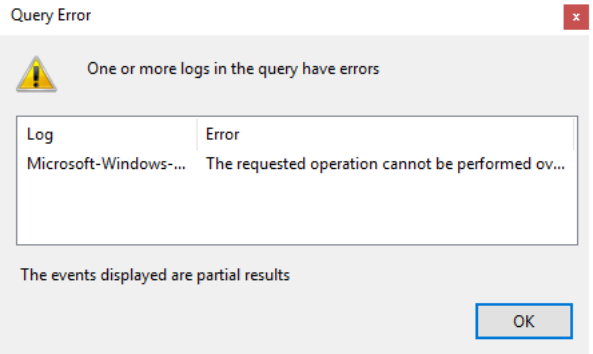


The analytic log must be configured as follows:

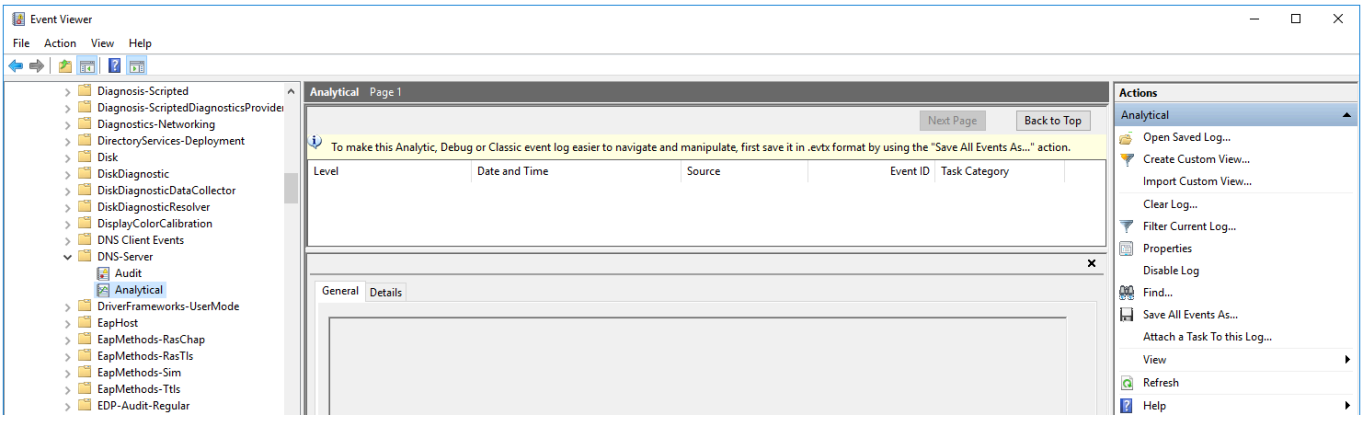


a. Click **Apply**, then click **OK**.

An error window is displayed.

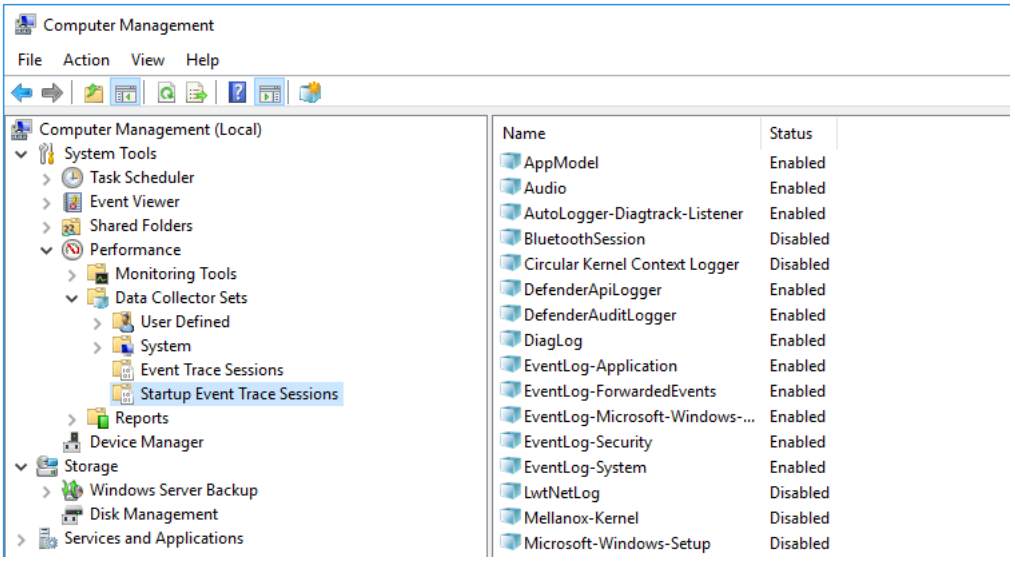


When analytic log rotation is enabled, events are not displayed. To view events, in the **Actions** pane, click **Stop logging**.



5. Start Computer management as administrator.

6. In the window that opens, go to the **System Tools** → **Performance** → **Startup Event Trace Sessions** folder.

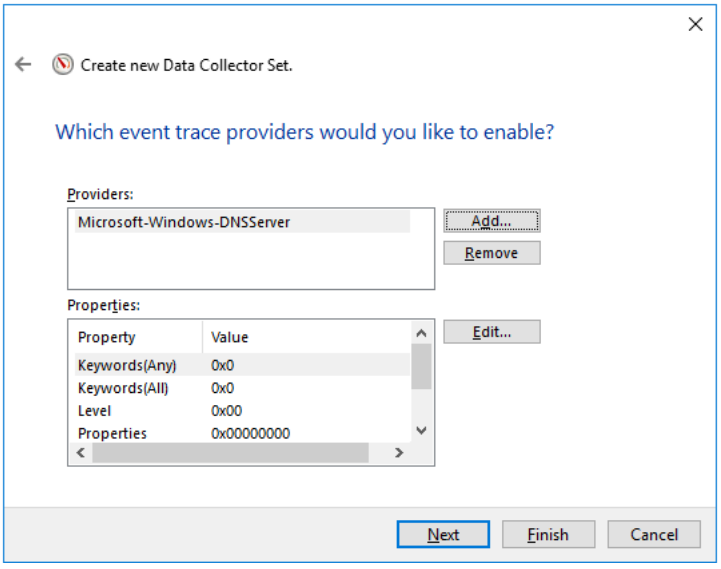


7. Create a provider:

   a. Open the context menu of the **Startup Event Trace Sessions** folder and select **Create** → **Data Collector Set**.

   b. In the window that opens, enter the name of the provider and click **Next**.

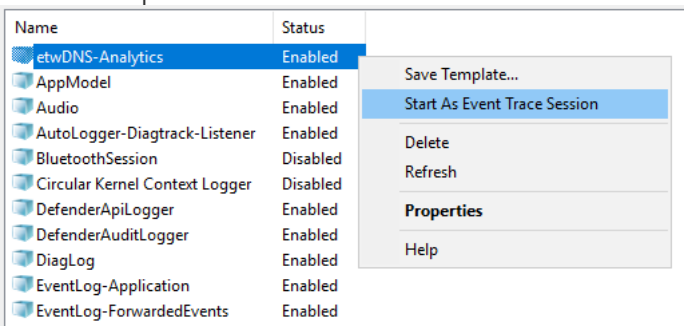
   c. Click **Add...** and in the displayed window, select the **Microsoft-Windows-DNSServer** provider.



The KUMA agent with the ETW connector works only with System.Provider.Guid: {EB79061A-A566-4698-9119-3ED2807060E7} - Microsoft-Windows-DNSServer.

a. Click **Next** twice, then click **Finish**.

8. Open the context menu of the created provider and select **Start As Event Trace Session**.

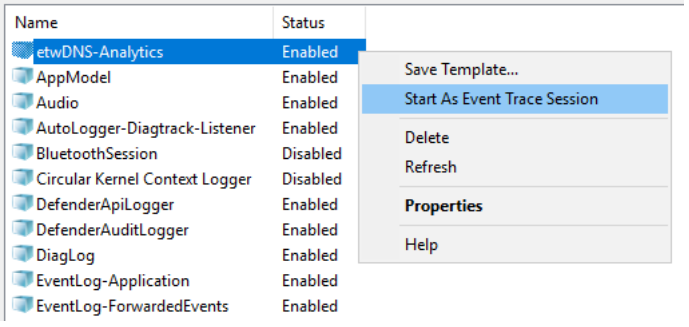


9. Go to the **Event Trace Sessions** folder.

   Event trace sessions are displayed.

10. Open the context menu of the created event trace session and select **Properties**.

11. In the window that opens, select the **Trace Sessions** tab and in the **Stream Mode** drop-down list, select **Real Time**.



12. Click **Apply**, then click **OK**.

DNS server event reception using the ETW connector is configured.

Configuring Windows event reception using Kaspersky Endpoint Security for Windows

In KES for Windows, starting from version 12.6, events can be sent from Windows logs to a Open Single Management Platform collector. In this way, Open Single Management Platform can get events from Windows logs (a limited set of EventIDs of Microsoft products is supported) from all hosts with KES for Windows 12.6 without installing Open Single Management Platform agents on such hosts. In KES for Windows versions 12.6–12.12, events with identical parameters are filtered: only the first event is sent, while subsequent similar events are not transferred. If the system generates a high volume of similar events, some of them may still be sent; this typically requires the occurrence of hundreds or thousands of such events. Events that differ in parameters (for example, when a different user logs in) are transferred without restrictions. The filtering logic is defined by AO Kaspersky Lab experts and is designed to reduce the volume of transferred events without losing the data necessary for attack detection.

To activate the functionality, you need:

- A valid Open Single Management Platform license
- KSC 14.2 or later
- KES for Windows version 12.6 or later

Configuring event receiving consists of the following steps:

1. Importing the normalizer into Open Single Management Platform.

   In Open Single Management Platform, you must configure getting updates through Kaspersky update servers.

   Click Import resources and in the list of normalizers available for installation, select **[OOTB] Microsoft Products via KES WIN**.

2. Creating a Open Single Management Platform collector for receiving Windows events.

   To receive Windows events, at the **Transport** step, select **TCP** or **UDP** and specify the port number that the collector must listen on. At the **Event parsing** step, select the **[OOTB] Microsoft Products via KES WIN** normalizer. At the **Event filtering** step, select the **[OOTB] Microsoft Products via KES WIN - Event filter for collector** filter.

3. Requesting a key from Open Single Management Platform Technical Support.

   If your license did not include a key for activating the functionality of sending Windows logs to the Open Single Management Platform collector, send the following message to Technical Support: `We have purchased a Open Single Management Platform license and are using KES for Windows version 12.6. We want to activate the functionality of sending Windows logs to the Open Single Management Platform collector. Please provide a key file to activate the functionality.` New Open Single Management Platform users do not need to make a Technical Support request because new users get 2 keys with licenses for Open Single Management Platform and for activating the KES for Windows functionality.

   In response to your message, you will get a key file.

4. Configuration on the side of KSC and KES for Windows.

   A key file that activates the functionality of sending Windows events to Open Single Management Platform collectors must be imported into KSC and distributed to KES endpoints in accordance with the instructions. You must also add Open Single Management Platform server addresses to the KES policy and specify network connection settings.

5. Verifying receipt of Windows events in the Open Single Management Platform collector

   You can verify that the Windows event source server is correctly configured in the Searching for related events section of the Open Single Management Platform web interface.

   Microsoft product events transmitted by KES for Windows are listed in the following table:

Event log	Event ID
DNS Server	150
DNS Server	770
MSExchange Management	1
Security	4781
Security	6416
Security	1100
Security	1102 / 517

Event log	Event ID
Security	1104
Security	1108
Security	4610 / 514
Security	4611
Security	4614 / 518
Security	4616 / 520
Security	4622
Security	4624 / 528 / 540
Security	4625 / 529
Security	4648 / 552
Security	4649
Security	4662
Security	4663
Security	4672 / 576
Security	4696
Security	4697 / 601
Security	4698 / 602
Security	4702
Security	4704 / 608
Security	4706
Security	4713/617
Security	4715
Security	4717 / 621
Security	4719 / 612
Security	4720 / 624
Security	4722 / 626
Security	4723 / 627
Security	4724 / 628
Security	4725 / 629
Security	4726 / 630
Security	4727
Security	4728 / 632
Security	4729 / 633
Security	4732 / 636
Security	4733 / 637
Security	4738 / 642
Security	4739/643
Security	4740 / 644
Security	4741
Security	4742 / 646
Security	4756 / 660

Event log	Event ID
Security	4757 / 661
Security	4765
Security	4766
Security	4767
Security	4768 / 672
Security	4769 / 673
Security	4770
Security	4771 / 675
Security	4775
Security	4776 / 680
Security	4778 / 682
Security	4780 / 684
Security	4794
Security	4798
Security	4817
Security	4876 / 4877
Security	4882
Security	4885
Security	4886
Security	4887
Security	4890
Security	4891
Security	4898
Security	4899
Security	4900
Security	4902
Security	4904
Security	4905
Security	4928
Security	4946
Security	4947
Security	4948
Security	4949
Security	4950
Security	4964
Security	5025
Security	5136
Security	5137
Security	5138
Security	5139
Security	5141

Event log	Event ID
Security	5142
Security	5143
Security	5144
Security	5145
Security	5148
Security	5155
Security	5376
Security	5377
Security	5632
Security	5888
Security	5889
Security	5890
Security	676
System	1
System	104
System	1056
System	12
System	13
System	6011
System	7040
System	7045
System, Source Netlogon	5723
System, Source Netlogon	5805
Terminal-Services-RemoteConnectionManager	1149
Terminal-Services-RemoteConnectionManager	1152
Terminal-Services-RemoteConnectionManager	20523
Terminal-Services-RemoteConnectionManager	258
Terminal-Services-RemoteConnectionManager	261
Windows PowerShell	400
Windows PowerShell	500
Windows PowerShell	501
Windows PowerShell	800
Application, Source ESENT	301
Application, Source ESENT	302
Application, Source ESENT	325
Application, Source ESENT	326
Application, Source ESENT	327
Application, Source ESENT	2001
Application, Source ESENT	2003
Application, Source ESENT	2005
Application, Source ESENT	2006

Event log	Event ID
Application, Source ESENT	216
Application	1000
Application	1002
Application	1 / 2

Configuring receipt of Codemaster Mirada events

You can configure the receipt of Codemaster Mirada events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring audit of the Codemaster Mirada system.
2. Creating a KUMA collector for receiving Codemaster Mirada events.

   To receive Codemaster Mirada, in the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] Codemaster Mirada syslog** normalizer, and at the **Transport** step, select the **tcp** or **udp** connector type.
3. Installing a collector in the KUMA network infrastructure.
4. Verifying receipt of Codemaster Mirada events in the KUMA collector

   You can verify that the Codemaster Mirada event source server is correctly configured in the **Events** section of the KUMA Console.

Configuring audit of the Codemaster Mirada system

The Codemaster Mirada system can send events over the Syslog protocol.

To configure event audit in the Codemaster Mirada system:

1. Connect to the Codemaster Mirada web interface under a user account that has administrative privileges.
2. Go to the **Settings** → **Syslog** section.
3. Enable the event transmission service using the toggle switch.
4. Select the type and format of the protocol by clicking the dot-in-a-circle icon.
5. In the **Host** field, specify the IP address of the KUMA collector.
6. In the **Protocol** field, specify the **UDP** or **TCP** transport protocol.
7. In the **Port** field, specify the port that the KUMA collector is listening on.

   The default port is 514.

8. In the **Format** field, specify the **RFC 3164** standard.

9. Click **Save** in the lower part of the page to save the changes.

Configuring receipt of Postfix events

You can configure the receipt of Postfix events in KUMA. Integration is only possible when sending events via syslog using the TCP protocol. The resources described in this article are available for KUMA 3.0 and newer versions.

Configuring event receiving consists of the following steps:

1. Configuring Postfix to send events.

2. Creating a KUMA collector for receiving Postfix events.

3. Verifying receipt of Postfix events in the KUMA collector

   You can verify that the Postfix event source server is correctly configured in the Searching for related events section of the KUMA Console.

The Postfix system generates events in two formats:

- Multi-line events containing information about messages (with a unique ID). These events have the following form:

  <syslog PRI> time host process_name: ID: information from base event 1

  <syslog PRI> time host process_name: id: info from base event 2

- Single-line events containing information about errors (without an ID). These events have the following form:

  <syslog PRI> time host process_name: severity: basic information for parsing

A set of KUMA resources is used to process Postfix events; this set of resources must be applied when creating a collector:

- Normalizer

- Aggregation rule

- Filters for destinations

The collector aggregates multi-line base events based on event ID, normalizes them, and sends the aggregated event to the storage and the correlator.

The aggregated event has the following form:

Service information from the aggregation rule: ID: information from base event 1, information from base event 2, information from base event n

After aggregation, the received event is sent to the same collector where the aggregated event is normalized.

Processing algorithm for Postfix events

The following algorithm was implemented to process Postfix events:

1. Initial normalization

   At this stage, initial normalization is performed for base events received via syslog that begin with the "<" character. The events are brought to a format suitable for subsequent aggregation: the first character is extracted from the event and put into the FlexString1 field, the identifier is put into the ExternalID field, and the host name is put into the DeviceHostName field. Basic normalization is performed in the main normalizer.

2. Checking for aggregation

   The event is examined to see if it is aggregated or not. As a result, non-aggregated events (the first character is not "{" and the ID is not empty) have an aggregation rule applied, and then aggregated events are sent for re-normalization.

3. Applying the aggregation rule

   At this stage, the aggregation rule is applied to the events, the base events are collated and take the following form:

   Service information from the aggregation rule: ID: information from base event 1, information from base event 2, information from base event n

4. After aggregation, the collated event is sent back to the same collector to subject the aggregated event to normalization.

   To close the event processing loop, you must specify the same collector as the destination. In the diagram, the destination is named "Loop" to draw attention to the event processing loop. You can give an arbitrary name to your destination.

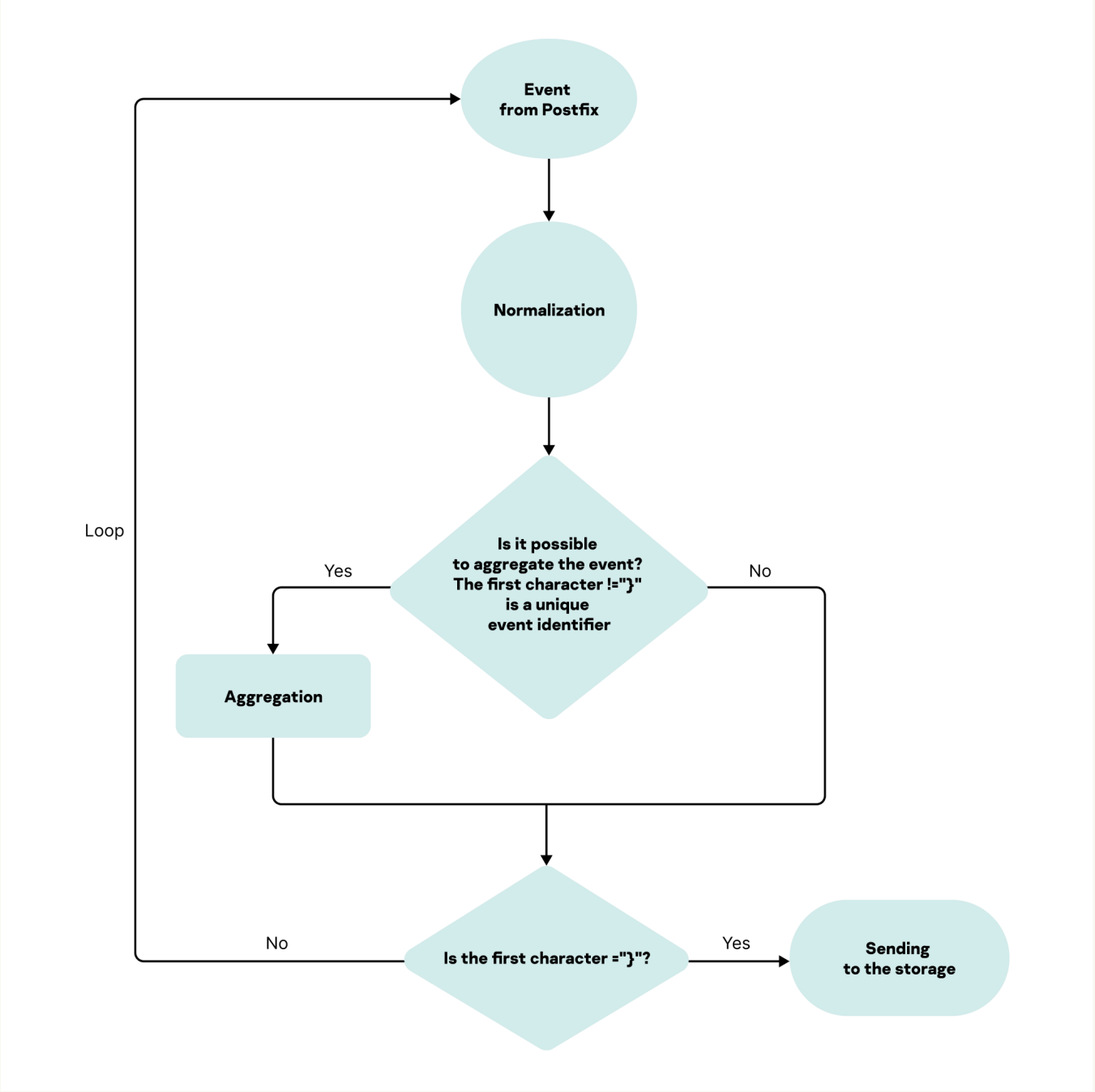
5. Normalization of the aggregated event

   Normalization of the aggregated event that begins with a "{" character is performed in the following extra normalizers: Aggregated events, Aggregated events. Message KV parser, Aggregated events. Message regex 1, Aggregated events. Message regex 2.

6. Sending to storage and the correlator

   Aggregated and normalized events are sent to storage and the correlator.

The following figure shows the flow chart of Postfix event processing.



Configuring Postfix to send events

By default, audit events of the Postfix system are output to /var/log/maillog or /var/log/mail.

To send events to KUMA:

1. Create a backup copy of the /etc/rsyslog.conf file.
2. Open the /etc/rsyslog.conf file for editing.
3. Add the following line to the end of the /etc/rsyslog.conf file:

   ```
   mail.* @@<IP address of the KUMA collector>:<port of the KUMA collector>
   ```

4. Save the /etc/rsyslog.conf file.

5. Restart the rsyslog service:

```
sudo systemctl restart rsyslog
```

Configuring a KUMA collector for receiving and processing Postfix events

To configure a KUMA collector for receiving Postfix events:

1. Import the [OOTB] Postfix package from the KUMA repository. The package is available for KUMA 3.0 and newer versions.

2. Create a new collector, and in the Collector Installation Wizard, configure the following:

    1. At the **Transport** step, in the **Type** field, select the **tcp** type, and in the **URL** field, specify the FQDN or IP address and port of the collector.

    2. At the **Event parsing** step, click **Add event parsing**, and in the displayed **Basic event parsing** window, in the **Normalizer** drop-down list, select the **[OOTB] Postfix syslog** normalizer.

    3. At the **Event aggregation** step, click **Add aggregation rule**, and in the displayed **Event aggregation** window, in the **Aggregation rule** drop-down list, select **[OOTB] Postfix. Aggregation rule**.

    4. At the **Routing** step, click **Add** and in the displayed **Create destination** window, create three destination points one by one—the same collector with the name `Loop`, a storage, and a correlator.

        a. Create a destination named `Loop` with the following parameters.

        - On the **Basic settings** tab, in the Type drop-down list, select the **tcp** transport type; in the **URL** field, specify the FQDN or IP address and port of the collector that you specified before at step 2.1 of these instructions.

        - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **Postfix. Filter for event aggregation** filter.

          This configuration is necessary to send the aggregated event to the same collector for subsequent normalization.

        b. Create a correlator destination:

        c. On the **Basic settings** tab, in the **Type** drop-down list, select **correlator** and fill in the **URL** field.

        d. On the **Advanced settings** tab, in the **Filter** drop-down list, select the **Postfix. Aggregated events to storage and correlator** filter.

        e. Create a storage destination:

        - On the **Basic settings** tab, in the **Type** drop-down list, select **storage** and fill in the **URL** field.

        - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **Postfix. Aggregated events to storage and correlator** filter.

          This configuration is necessary to send the aggregated normalized event to storage and the correlator.

3. Click the **Create** button.

   The collector service is created with the settings specified in the KUMA Console. The command for installing the service on the server is displayed.

4. Copy the collector installation command and run it on the relevant server.

The collector is configured to receive and process Postfix events.

Configuring receipt of CommuniGate Pro events

You can configure the receipt of CommuniGate Pro 6.1 events in KUMA. Integration is only possible when sending events via syslog using the TCP protocol. The resources described in this article are available for KUMA 3.0 and newer versions. Processing of SIP module events is supported (such events contain the "SIPDATA" character sequence).

Configuring event receiving consists of the following steps:

1. Configuring CommuniGate Pro to send events

2. Configuring the KUMA collector for receiving CommuniGate Pro events

3. Verifying receipt of CommuniGate Pro events in the KUMA collector

   You can verify that the CommuniGate Pro event source server is correctly configured in the Searching for related events section of the KUMA Console.

The CommuniGate Pro system generates an audit event as several separate records that look like this:

```
<event code> timestamp ID direction: information from base event 1

<event code> timestamp ID direction: information from base event 2

<event code> timestamp ID direction: base information n
```

A set of KUMA resources is used to process CommuniGate Pro events; this set of resources must be applied when creating a collector:

- Normalizer

- Aggregation rule

- Filters for destinations

The collector aggregates multi-line base events based on event ID, normalizes them, and sends the aggregated event to the storage and the correlator.

The aggregated event has the following form:

```
Service information from the aggregation rule: ID: information from base event 1,
information from base event 2, information from base event n
```

After aggregation, the received event is sent to the same collector where the aggregated event is normalized.

Processing algorithm for CommuniGate Pro events

The following algorithm was implemented to process CommuniGate Pro events:

1. Initial normalization

   At this stage, the initial normalization of base events is performed. The first character in the base event is a numeral. The events are brought to a format suitable for subsequent aggregation: the first character is extracted from the event and put into the DeviceCustomString1 field, the identifier is put into the ExternalID field, and the host name is put into the DeviceHostName field. Basic normalization is performed in the main normalizer.

2. Checking for aggregation

   The event is examined to see if it is aggregated or not. As a result, non-aggregated events (the first character is a numeral) have an aggregation rule applied, and then aggregated events are sent for re-normalization. Aggregation is performed using the **[OOTB] CommuniGate Pro. Aggregation rule**.

3. Applying the aggregation rule

   At this stage, the aggregation rule is applied to the events, the base events are collated and take the following form:

   Service information from the aggregation rule: ID: information from base event 1, information from base event 2, information from base event n

   After aggregation, the collated event is sent back to the same collector to subject the aggregated event to normalization.

   To close the event processing loop, you must specify the same collector as the destination. In the diagram, the destination is named "Loop" to draw attention to the event processing loop. You can give an arbitrary name to your destination.

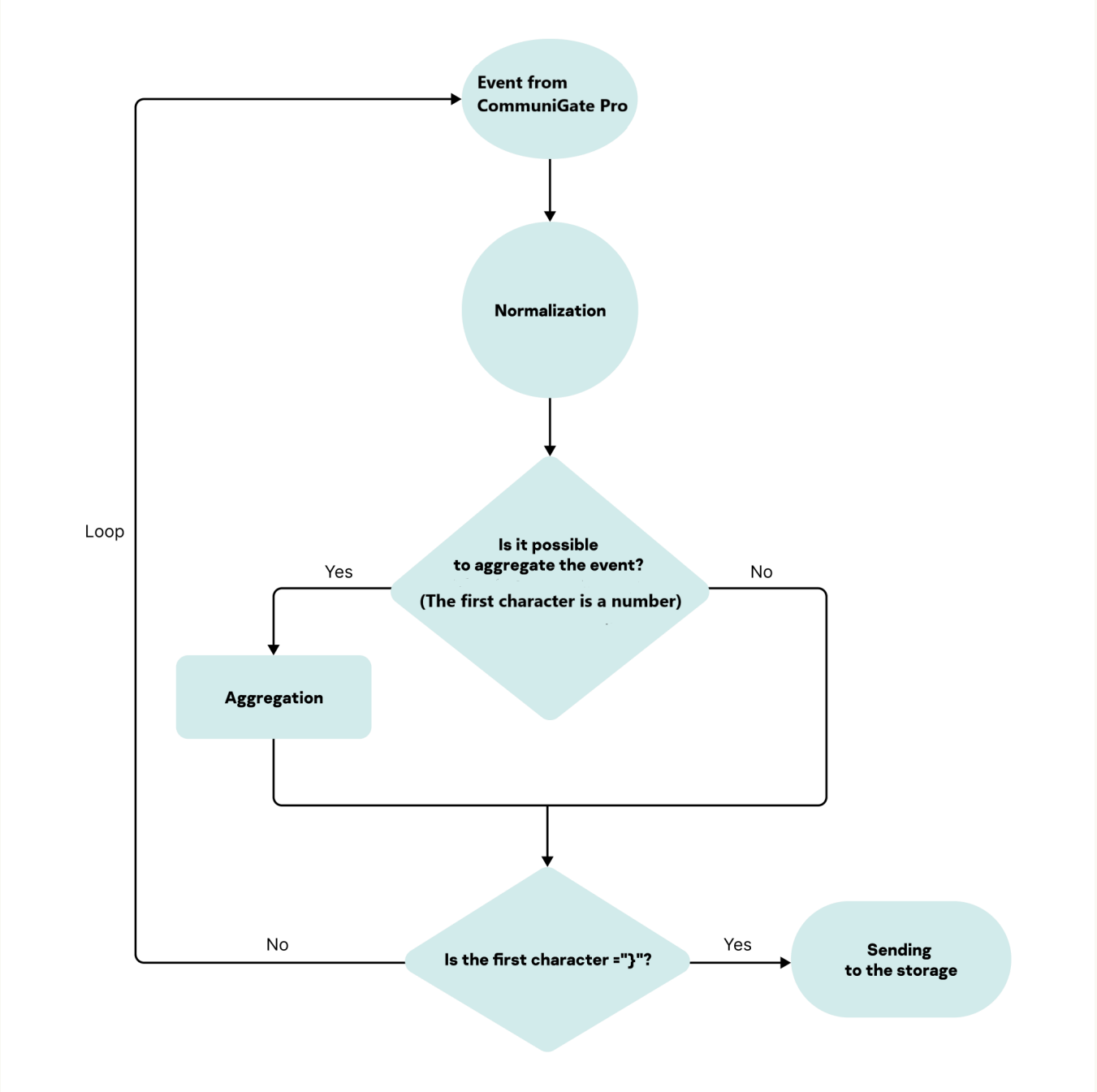
4. Normalization of the aggregated event

   Normalization of the aggregated event that begins with a "{" character is performed in the following extra normalizers: Aggregated events, Aggregated events - kv part.

5. Sending to storage and the correlator

   Aggregated and normalized events are sent to storage and the correlator.

The following figure shows the flow chart of CommuniGate Pro event processing.



Configuring CommuniGate Pro to send events

By default, CommuniGate Pro audit events are sent to .log files in the /var/CommuniGate/SystemLogs/ directory.

To send events to KUMA, you need to install the KUMA agent on the CommuniGate Pro server and configure it to read .log in the /var/CommuniGate/SystemLogs/ directory and send them to the KUMA collector over TCP.

To create an agent that will read and send events to KUMA:

1. In the KUMA Console, go to **Resources and services** → **Agents** and click **Add**.

2. This opens the **Create agent** window; in that window, on the **Basic settings** tab, in the **Name** field, specify the agent name.

3. On the **Config #1** tab, fill in the following fields:

   a. In the **Connector** group of settings on the **Basic settings** tab, set the following values for the connector:

      1. In the **Name** field, enter a name, for example, `CommuniGate file`.

      2. In the **Type** drop-down list, select **file**.

      3. In the **File path** field, enter the following value:
         /var/CommuniGate/SystemLogs/.*.log

   b. In the **Destinations** group of settings on the **Basic settings** tab, set the following values for the destination:

      1. In the **Name** field, enter a name, for example, `CommuniGate TCP collector`.

      2. In the **Type** drop-down list, select **tcp**.

      3. In the **URL** field, enter the FQDN or IP address and port of the KUMA collector.

4. Click the **Create** button.

5. When the agent service is created in KUMA, install the agent on the network infrastructure devices from which you want to send data to the collector.

Configuring a KUMA collector for receiving and processing CommuniGate Pro events

To configure a KUMA collector for receiving CommuniGate Pro events:

1. Import the [OOTB] CommuniGate Pro package from the KUMA repository. The package is available for KUMA 3.0 and newer versions.

2. Create a new collector, and in the Collector Installation Wizard, configure the following:

   1. At the **Transport** step, in the **Type** field, select the **tcp** type, and in the **URL** field, specify the FQDN or IP address and port of the collector.

   2. At the **Event parsing** step, click **Add event parsing**, and in the displayed **Basic event parsing** window, in the **Normalizer** drop-down list, select the **[OOTB] CommuniGate Pro** normalizer.

   3. At the **Event aggregation** step, click **Add aggregation rule**, and in the displayed **Event aggregation** window, in the **Aggregation rule** drop-down list, select **[OOTB] CommuniGate Pro. Aggregation rule**.

4. At the **Routing** step, click **Add** and in the displayed **Create destination** window, create three destination points one by one—the same collector with the name `Loop`, a storage, and a correlator.

   a. Create a destination named `Loop` with the following parameters:

      - On the **Basic settings** tab, in the **Type** drop-down list, select the **tcp** transport type; in the **URL** field, specify the FQDN or IP address and port of the collector that you specified before at step 2.1 of these instructions.

      - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **[OOTB] CommuniGate Pro. Filter for event aggregation** filter.

      This configuration is necessary to send the aggregated event to the same collector for subsequent normalization.

   b. Create a correlator destination:

      - On the **Basic settings** tab, in the **Type** drop-down list, select **correlator** and fill in the **URL** field.

      - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **[OOTB] CommuniGate Pro. Aggregated events to storage and correlator** filter.

   c. Create a storage destination:

      - On the **Basic settings** tab, in the **Type** drop-down list, select **storage** and fill in the **URL** field.

      - On the **Advanced settings** tab, in the **Filter** drop-down list, select the **[OOTB] CommuniGate Pro. Aggregated events to storage and correlator** filter.

      This configuration is necessary to send the aggregated normalized event to storage and the correlator.

3. Click the **Create** button.

   The collector service is created with the settings specified in the KUMA Console. The command for installing the service on the server is displayed.

4. Copy the collector installation command and run it on the relevant server.

The collector is configured to receive and process CommuniGate Pro events.

Configuring receipt of Yandex Cloud events

You can configure the receipt of Yandex Cloud events in KUMA. The normalizer supports processing configuration-level audit events stored in .json files.

Configuring event receiving consists of the following steps:

1. Configuring audit of Yandex Cloud events.

2. Configuring export of Yandex Cloud events.

3. Configuring a KUMA collector for receiving and processing Yandex Cloud events.

To receive Yandex Cloud events in the KUMA Collector Installation Wizard:

1. In the KUMA Collector Installation Wizard, at the **Transport** step, select the connector of the **file** type.
2. In the **URL** field, enter `/var/log/yandex-cloud/<audit_trail_id>/*/*/*/*.json`, where <audit_trail_id> is the ID of the audit.
3. At the **Event parsing** step, in the **Normalizer** field, select **[OOTB] Yandex Cloud**.

4. Installing a collector in the KUMA network infrastructure.
5. Verifying receipt of Yandex Cloud events in the KUMA collector

   To verify that the Yandex Cloud event source server is configured correctly, you can search for related events.

Configuring audit of Yandex Cloud events

Configuring event export proceeds in stages:

1. Preparing the environment for working with Yandex Cloud.
2. Creating a bucket for audit logs.
3. Creating an encryption key in the Key Management Service.
4. Enabling bucket encryption.
5. Creating service accounts.
6. Creating a static key.
7. Assigning roles to service accounts.
8. Creating an audit trail.

Preparing the environment for working with Yandex Cloud.

To manage the configuration, you need Yandex Cloud CLI; install and initialize it.

Note: by default, audit is performed in the Yandex Cloud folder specified in the CLI profile. You can specify a different folder using the `--folder-name` or `--folder-id` parameter.

To configure the audit, you need an active billing account because a fee is charged for using the Yandex Cloud infrastructure.

To configure Yandex Cloud audit, you need an active billing account:

1. Go to the management console, then log in to Yandex Cloud or register.
2. On the Yandex Cloud Billing page, make sure that you have a billing account connected and that it has the ACTIVE or TRIAL_ACTIVE status. If you do not have a billing account, create one.

If you have an active billing account, you can create or select a Yandex Cloud folder in which your infrastructure will work, on the cloud page.

Creating a bucket for audit logs

To create a bucket:

1. In the management console, go to the folder in which you want to create the bucket, for example, **example-folder**.
2. Select the **Object Storage** service.
3. Click **Create bucket**.
4. On the bucket creation page:
    1. Enter the bucket name in accordance with the naming rules, for example **kumabucket**.
    2. If necessary, limit the maximum size of the bucket. Size 0 means no limit and is equivalent to the enabled **No limit** option.
    3. As the access type, select **Restricted**.
    4. Select the default storage class.
    5. Click **Create bucket**.

The bucket is created.

Creating an encryption key in the Key Management Service

To create an encryption key:

1. In the management console, go to the **example-folder** folder.
2. Select the **Key Management Service**.
3. Click the **Create key** button and specify the following settings:
    - Name (for example, **kuma-kms**).
    - Encryption algorithm, **AES-256**.
    - Keep default values for the rest of the settings.
4. Click **Create**.

The encryption key is created.

Enabling bucket encryption

To enable bucket encryption:

1. In the management console, go to the bucket you created earlier.
2. In the left pane, select **Encryption**.
3. In the **KMS key** field, select the **kuma-kms** key.
4. Click **Save**.

Bucket encryption is enabled.

Creating service accounts

To create service accounts (a separate account for the trail and a separate account for the bucket):

1. Create the **sa-kuma** service account:
    1. In the management console, go to the **example-folder** folder.
    2. In the upper part of the screen, go to the **Service accounts** tab.
    3. Click **Create service account** and enter the name of the service account, for example, **sa-kuma**, making sure the name complies with the naming rules:
        - length: 3 to 63 characters
        - may contain lower-case letters of the Latin alphabet, numerals, and hyphens
        - the first character must be a letter, the last character may not be a hyphen
    4. Click **Create**.
2. Create the **sa-kuma-bucket** service account:
    1. In the management console, go to the **example-folder** folder.
    2. In the upper part of the screen, go to the **Service accounts** tab.
    3. Click **Create service account** and enter the name of the service account, for example, **sa-kuma-bucket**, making sure the name complies with the naming rules:
        - length: 3 to 63 characters
        - may contain lower-case letters of the Latin alphabet, numerals, and hyphens
        - the first character must be a letter, the last character may not be a hyphen
    4. Click **Create**.

The service accounts are created.

Creating a static key

You will need the key ID and the private key when mounting the bucket. You can create a key using the management console or the CLI.

To create a key using the management console:

1. In the management console, go to the **example-folder** folder.
2. In the upper part of the screen, go to the **Service accounts** tab.
3. Select the **sa-kuma-bucket** service account and click the row with its name.
4. In the upper panel, click **Create new key**.
5. Select **Create static access key**.
6. Enter a description for the key and click **Create**.
7. Save the ID and the secret key.

   The static access key is created. The key value will become unavailable when you close the dialog.

To create a key using the CLI:

1. Create an access key for the **sa-kuma-bucket** service account:

   ```
   yc iam access-key create --service-account-name sa-kuma-bucket
   ```

   Result:

   ```
   access_key:
   id: aje*******k2u
   service_account_id: aje*******usm
   created_at: "2022-09-22T14:37:51Z"
   key_id: 0n8*******0YQ
   secret: JyT*******zMP1
   ```

2. Save the key_id and the key from the 'secret' value. You will not be able to get the key value again.

   The access key is created.

Assigning roles to service accounts

To assign the audit-trails.viewer, storage.uploader, and kms.keys.encrypterDecrypter roles to the sa-kuma service account:

1. In the CLI, assign the audit-trails.viewer role to the folder:

   ```
   yc resource-manager folder add-access-binding \
   --role audit-trails.viewer \
   --id <folder_id> \
   --service-account-id <service_account_id>
   ```

Where:

- `--role` is the assigned role.
- `--id` is the ID of the 'example-folder' folder.
- `--service-account-id` is the ID of the sa-kuma service account.

2. Assign the storage.uploader role to the folder with the bucket:

```
yc resource-manager folder add-access-binding \
--role storage.uploader \
--id <folder_id> \
--service-account-id <service_account_id>
```

Where:

- `--role` is the assigned role.
- `--id` is the ID of the 'example-folder' folder.
- `--service-account-id` is the ID of the sa-kuma service account.

3. Assign the kms.keys.encrypterDecrypter role to the kuma-kms encryption key:

```
yc kms symmetric-key add-access-binding \
--role kms.keys.encrypterDecrypter \
--id <key_id> \
--service-account-id <service_account_id>
```

Where:

- `--role` is the assigned role.
- `--id` is the ID of the kuma-kms KMS key.
- `--service-account-id` is the ID of the sa-kuma service account.

To assign the storage.viewer and kms.keys.encrypterDecrypter roles to the sa-kuma-bucket service account:

1. In the CLI, assign the storage.viewer role to the folder:

```
yc resource-manager folder add-access-binding \
--id <folder_id> \
--role storage.viewer \
--service-account-id <service_account_id>
```

Where:

- `--id` is the ID of the 'example-folder' folder.
- `--role` is the assigned role.
- `--service-account-id` is the ID of the sa-kuma-bucket service account.

2. Assign the kms.keys.encrypterDecrypter role to the kuma-kms encryption key:

```
yc kms symmetric-key add-access-binding \
--role kms.keys.encrypterDecrypter \
--id <key_id> \
--service-account-id <service_account_id>
```

Where:

- `--role` is the assigned role.
- `--id` is the ID of the kuma-kms KMS key.
- `--service-account-id` is the ID of the sa-kuma-bucket service account.

Creating an audit trail

To create an audit trail:

1. In the management console, go to the **example-folder** folder.
2. Select the **Audit Trails** service.
3. Click **Create trail** and specify a name for the trail you are creating, for example, **kuma-trail**.
4. In the **Destination** section, specify the parameters of the destination object:
   - **Destination**: **Object Storage**.
   - **Bucket**: The name of the bucket, for example **kumabucket**.
   - **Object prefix**: Optional parameter used in the full name of the audit log file.

     Use a prefix if you store audit logs and third-party data in the same bucket. Do not use the same prefix for logs and other objects in the bucket because this may cause logs and third-party objects to overwrite each other.
   - **Encryption key**: specify the kuma-kms encryption key that the bucket is encrypted with.
5. In the **Service account** section, select **sa-kuma**.
6. In the **Collecting management events** section, specify the settings for collecting management events audit logs:
   - **Collecting events**: Select **Enabled**.
   - **Resource**: Select **Folder**.
   - **Folder**: Does not require filling, contains the name of the current folder.
7. In the **Collecting data events**, in the **Collecting events** field, select **Disabled**.
8. Click **Create**.

Configuring export of Yandex Cloud events

The bucket must be mounted on the server on which the KUMA collector will be installed.

To mount the bucket:

1. On the server, create a directory for the 'kuma' user:

   ```
   sudo mkdir /home/kuma
   ```

2. On the server, create a file with a static access key for the sa-kuma-bucket service account and grant appropriate access permissions to the 'kuma' user:

   ```
   sudo bash -c 'echo <access_key_ID>:<secret_access_key> > /home/kuma/.passwd-s3fs'
   sudo chmod 600 /home/kuma/.passwd-s3fs
   sudo chown -R kuma:kuma /home/kuma
   ```

3. Install the s3fs package:

   ```
   sudo apt install s3fs
   ```

4. Create a directory where the bucket must be mounted and grant permissions to the kuma user:

   ```
   sudo mkdir /var/log/yandex-cloud/
   sudo chown kuma:kuma /var/log/yandex-cloud/
   ```

5. Mount the bucket:

   ```
   sudo s3fs kumabucket /var/log/yandex-cloud -o passwd_file=/home/kuma/.passwd-s3fs -o
   url=https://storage.yandexcloud.net -o use_path_request_style -o uid=$(id -u kuma) -o
   gid=$(id -g kuma)
   ```

   You can configure the bucket to be mounted at operating system startup by adding a line to /etc/fstab, for example:

   ```
   s3fs#kumabucket /var/log/yandex-cloud fuse _netdev,uid=<kuma_uid>,gid=
   <kuma_gid>,use_path_request_style,url=https://storage.yandexcloud.net,passwd_file=/home
   s3fs 0 0
   ```

   Where:

   <kuma_uid> is the ID of `kuma` user in the operating system.

   <kuma_gid> is the ID of `kuma` group of users in the operating system.

   To find out the kuma_uid and kuma_gid, run the following command in the console:

   ```
   id kuma
   ```

6. Verify that the bucket is mounted:

   ```
   sudo ls /var/log/yandex-cloud/
   ```

   If everything is configured correctly, the command returns <audit_trail_id>, where <audit_trail_id> is the audit trail ID.

Export of Yandex Cloud events is configured. Events will be located in directories in .json files:

/var/log/yandex-cloud/{audit_trail_id}/{year}/{month}/{day}/*.json

Configuring receipt of Microsoft 365 events

You can configure the receipt of events from the Microsoft 365 (Office 365) cloud solution in KUMA.

Configuring event receiving consists of the following steps:

1. Configuring access to Office 365 management APIs using standard Microsoft methods

   To receive events in KUMA, grant the necessary set of API permissions:

   Microsoft.Graph

   Directory.Read.All

   Office 365 management API

   ActivityFeed.Read

   ActivityFeed.Read.Dlp

2. Creating a KUMA collector

   To receive Microsoft 365 events, create a collector with the following parameters:

   - At the **Transport** step, specify the **office365** connector type.
   - At the **Parsing events** step, specify the **[OOTB] Microsoft Office 365 json** normalizer.

3. Installing a collector in a KUMA network infrastructure

4. Verifying receipt of Windows Microsoft 365 in the KUMA collector

   To verify that the Microsoft 365 event source server is configured correctly, you can search for related events.

Configuring receipt of the Kontinent encryption system events

The Open Single Management Platform distribution kit includes the kuma-kont utility, which consists of an executable file and a configuration file. Supported operating systems include Astra Linux, Ubuntu 22.04 LTS, and RED OS 7.3.4 and 8. This utility allows lets you connect to the MSSQL database, receive events of the Kontinent encryption system, and send data to the Open Single Management Platform collector in JSON format. Data import is supported for Kontinent 3.9.2.

Configuring the retrieval of Kontinent events from an MSSQL database and sending them to the Open Single Management Platform collector involves the following steps:

1. Preparing to receive Kontinent events.

   At this step, you need to create a KUMA collector that will receive events, and also create a user in the MSSQL database.

2. Configuring the kuma-kont utility.

   At this step, you need to create the configuration file of the utility and the service of the utility.

3. Starting the kuma-kont utility.

   After activation, the utility starts exporting events from the MSSQL database and sending them to the Open Single Management Platform collector. It keeps a log in the kuma-kont.log file.

Preparing to receive Kontinent events

1. Creating a KUMA collector.

   1. Creating a KUMA collector for Kontinent 3.9 events.

      To receive audit events of the Kontinent 3.9 system using the Collector Installation Wizard, at the **Event parsing** step, select the **[OOTB] SecurityCode Continent 3.9 json** normalizer. Also, on the **Transport** tab, you need to specify the port and protocol in accordance with paragraph 2.1.4 of the *Configuring the export of Kontinent 3.9 events* instructions.

   2. Installing a collector in the Open Single Management Platform network infrastructure.

2. Creating an account in the MS SQL DBMS.

   Create an account in the database with the minimum necessary rights to connect and read data from the following tables: ALERTLOG, SERVERACCESSLOG, SYSTEMLOG, PACKETLOG, FILTERS.

3. Ensuring network connectivity.

   You must ensure network connectivity between the server on which the utility is running and the database server of the Kontinent system, as well as between the server on which the utility is running and the Open Single Management Platform collector server.

Configuring the kuma-kont utility

Configuring the kuma-kont utility involves the following steps:

1. Creating the kuma-kont-config.yaml configuration file.

2. Creating the kuma-kont service.

Creating the kuma-kont-config.yaml configuration file

To create the kuma-kont-config.yaml configuration file:

1. Go to the Open Single Management Platform installer directory by executing the following command:

   ```
   cd kuma-ansible-installer/roles/kuma/files
   ```

2. Extract the kuma-kont.tar archive:

   ```
   tar -xvf kuma-kont.tar
   ```

3. Go to the directory of the utility:

   ```
   cd kuma-kont
   ```

4. Copy the kuma-kont-config-template.yaml template to create a configuration file named kuma-kont-config.yaml:

   ```
   cp kuma-kont-config-template.yaml kuma-kont-config.yaml
   ```

5. Change read permissions for the configuration file:

```
sudo chown <user name>:<group> kuma-kont-config.yaml
sudo chmod 600 kuma-kont-config.yaml
```

Replace <user name> with the operating system user name that the service will use.

6. Edit the settings in the kuma-kont-config.yaml configuration file.

   Specify the values of the following settings:

   - General settings: state_file, log_file_path.
   - Kontinent Database: name, user, password, host, port.
   - Open Single Management Platform Collector: address, port, protocol.

     An example of configuration is given in the kuma-kont-config-template.yaml file.

7. Save the changes to the file.

The configuration file is created.

Creating the service

To create the kuma-kont service:

1. Change the execution permissions for the executable file:

   ```
   sudo chown <user name>:<group> <path to executable file>
   sudo chmod 700 <path to executable file>
   ```

2. Create the /etc/systemd/system/kuma-kont.service file with the following contents:

   ```
   [Unit]
   Description=Connector Kontinent 3.9
   After=network.target
   [Service]
   User=<user name>
   ExecStart=<path to executable file> --config <path to configuration file>
   Restart=always
   RestartSec=10
   [Install]
   WantedBy=multi-user.target
   ```

   Replace <user name> with the user name of the operating system user that the service will use.

3. Specify the path to the executable file.

   Specify the path to the kuma-kont executable file in the ExecStart field.

   For example, if the kuma-kont file is located in the /usr/bin/ directory, leave the default value.

4. Specify the path to the configuration file.

   Specify the path to the kuma-kont-config.yaml configuration file in the ExecStart field. For example, if the kuma-kont-config.yaml file is located in the /path/to/ directory, specify that path.

   Example kuma-kont.service file

   ```
   [Unit]
   Description=Continent Service
   After=network.target
   [Service]
   ```

```
User=user-example
ExecStart=./usr/bin/kuma-kont --config /path/to/kuma-kont-config.yaml
Restart=always
RestartSec=10
[Install]
WantedBy=multi-user.target
```

5. Load the systemd configuration:

   ```
   sudo systemctl daemon-reload
   ```

6. Make the service run at startup:

   ```
   sudo systemctl enable kuma-kont
   ```

The kuma-kont utility will be started as a service at system startup.

Starting and managing the kuma-kont service

To start the utility service, run the following command:

```
sudo systemctl start kuma-kont
```

The collection of events from the MSSQL database begins and the events are sent to the Open Single Management Platform collector.

The utility service can be controlled using the following commands:

1. Start the kuma-kont service:

   ```
   sudo systemctl start kuma-kont
   ```

2. Stop the kuma-kont service:

   ```
   sudo systemctl stop kuma-kont
   ```

3. Restart the kuma-kont service:

   ```
   sudo systemctl restart kuma-kont
   ```

You can also send events using the kuma-kont utility running as an application. To start sending events to the Open Single Management Platform collector, you need to run the following command while in the directory with the executable file:

```
./kuma-kont --config <path to the kuma-kont-config.yaml file>
```

Behavior when connection is lost

Connection to the MSSQL database.

If the connection to the MSSQL database is lost, the utility tries to reconnect to it every n seconds (where n is equal to the value of the poll_interval setting in the configuration file) until the connection is restored.

Connection to the Open Single Management Platform collector

If the TCP protocol is used to send events, when the connection to the Open Single Management Platform collector is lost, the utility also attempts to restore it. The interval between connection attempts increases until it reaches one hour, and subsequent reconnection attempts are made once per hour.

Known limitations

The event from the `AlertLog` table with AL_CATEGORY=5 is not fully decoded.

Settings in the kuma-kont-config.yaml configuration file

The table lists the settings that you can specify in the kuma-kont-config.yaml file.

Settings in the kuma-kont-config.yaml configuration file

Setting	Description
`debug`	Optional setting in the **General settings** group. Used when debug level log output is required. If false, only start and stop messages are included, as well as errors that occurred during the sending process. Available values: • **true** • **false** Default setting: **false**.
`need_reconnect`	An optional setting in the **General settings** group. This setting determines whether the utility reconnects to the database to make each request. Available values: • **true** • **false** Default setting: **false**.
`state_file`	Required setting in the **General settings** group. Absolute path to the JSON file in which the state of the processed logs is saved. The file is created automatically at this path. Make sure the user has sufficient permissions to write to this directory. Example: /home/user/kuma-kont/state.json
`log_file_path`	Required setting in the **General settings** group. Absolute path to the log file for recording connector activity. The file is created automatically at this path. Make sure the user has sufficient permissions to write to this directory. Example: /home/user/kuma-kont/kuma-kont.log
`poll_interval`	An optional setting in the **General settings** group. Database polling interval. The default value is 10 seconds. For high EPS, we recommend a value less than 15 seconds.
`name`	Required setting in the **Kontinent Database** group. Name of the database in which Kontinent logs are received.
`user`	Required setting in the **Kontinent Database** group. Name of the database user.
`password`	Required setting in the **Kontinent Database** group. Password of the database user.
`host`	Required setting in the **Kontinent Database** group. URL of the database server.

Setting	Description
`port`	Required setting in the **Kontinent Database** group. Port of the database server. Make sure it is accessible from the server where the utility is installed.
`limit`	Optional setting in the **Kontinent Database** group. Limit on the number of events that can be downloaded from the database as part of one poll. By default, 100,000 events per database poll. If EPS is high, we recommend a value equal or greater than this.
`alert_log`	Optional setting in the **Last Id** group. Specifies the event of the AlertLog type from which you want to start sending events to the Open Single Management Platform collector. Only works on first run or when the state file does not exit. The default value is 0.
`packet_log`	Optional setting in the **Last Id** group. Specifies the event of the PacketLog type from which you want to start sending events to the Open Single Management Platform collector. Only works on first run or when the state file does not exit. The default value is 0.
`server_access_log`	Optional setting in the **Last Id** group. Specifies the event of the ServerAccessLog type from which you want to start sending events to the Open Single Management Platform collector. Only works on first run or when the state file does not exit. The default value is 0.
`system_log`	Optional setting in the **Last Id** group. Specifies the event of the SystemLog type from which you want to start sending events to the Open Single Management Platform collector. Only works on first run or when the state file does not exit. The default value is 0.
`address`	Required setting in the **Open Single Management Platform Collector** group. Fully qualified domain name (FQDN) or IP address of the server where the Open Single Management Platform collector is installed.
`port`	Required setting in the **Open Single Management Platform Collector** group. Port of the Open Single Management Platform collector.
`protocol`	Required setting in the **Open Single Management Platform Collector** group. Data transfer protocol used by the Open Single Management Platform collector. If the udp protocol is used, events are not resent if connection to the Open Single Management Platform collector is lost. Available values: • **tcp** • **udp**

Integration with other solutions

In this section, you'll learn how to integrate KUMA with other solutions to enrich its functionality.

Integration with Kaspersky Security Center

You can create or edit Kaspersky Security Center integration settings in OSMP Console.

In OSMP Console, you can view the integration with selected Kaspersky Security Center Servers for one, several, or all KUMA tenants. If integration with Kaspersky Security Center is enabled, you can manually import assets, edit the automatic scheduled import interval, or temporarily disable scheduled import.

Configuring the data refresh interval for Kaspersky Security Center assets

KUMA imports the following device information received from Kaspersky Security Center Network Agents:

- Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
- Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To configure the data refresh interval for asset data from Kaspersky Security Center:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant into which you want to import KSC assets information.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **KSC import** tab, make sure that the **State** toggle switch is turned on.
5. Specify intervals for automatic import of asset information and asset vulnerability information from Kaspersky Security Center:
   - Under **Hardware information**, in the **Import interval (hours)** field, enter the interval in hours for the automatic import of information about the basic attributes of assets. The default setting is 1 hour.
   - Under **Assets attributes**, in the **Import interval (hours)** field, enter the interval in hours for automatic import of information about other attributes of assets. The default setting is 12 hours.

     Importing the information about asset attributes (vulnerabilities, software, owners) may involve downloading a large amount of data, which may take a longer time to complete, we recommend setting a longer interval than for the hardware information import.
6. Click the **Save** button.

Kaspersky Security Center asset data update settings for the selected tenant are configured.

If the tenant you want is missing from the list of tenants, use the OSMP console to add it to the list of tenants.

Manual import of Kaspersky Security Center assets

KUMA imports the following device information received from Kaspersky Security Center Network Agents:

- Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
- Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To manually import Kaspersky Security Center assets:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant into which you want to import KSC assets information.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **KSC import** tab, make sure that the **State** toggle switch is turned on.
5. Specify intervals for automatic import of asset information and asset vulnerability information from Kaspersky Security Center:

   - To import information about the basic attributes of assets, under **Hardware information**, click the **Import** button.
   - To import information about other attributes of assets, under **Assets attributes**, click the **Import** button.

The *KSC assets import* task is created. Kaspersky Security Center assets for the specified tenant are imported regardless of the configured schedule.

Importing events from the Kaspersky Security Center database

In KUMA, you can receive events from the Kaspersky Security Center SQL database. Events are received using the collector, which uses the following resources:

- Predefined connector: [OOTB] KSC MSSQL or [OOTB] KSC MySQL.
- Predefined [OOTB] KSC from SQL normalizer.

Configuring the import of events from Kaspersky Security Center involves the following steps:

1. Create a copy of the predefined connector.

   The settings of the predefined connector are not editable, therefore, to configure the connection to the database server, you must create a copy of the predefined connector.
2. Creating a collector:

   - In the web interface.
   - On the server.

To configure the import of events from Kaspersky Security Center:

1. Create a copy of the predefined connector corresponding to the type of database used by Kaspersky Security Center:

   a. In OSMP Console, in the **Monitoring** → **Resources** → **Connectors** section, find the relevant predefined connector in the folder hierarchy, select the check box next to that connector, and click **Duplicate**.

   b. This opens the **Create connector** window; in that window, on the **Basic settings** tab, in the **Default query** field, if necessary, replace the KAV database name with the name of the Kaspersky Security Center database you are using.

   An example of a query to the Kaspersky Security Center SQL database

```
SELECT ev.event_id AS externalId, ev.severity AS severity, ev.task_display_name AS taskDisplayName,

      ev.product_name AS product_name, ev.product_version AS product_version,

       ev.event_type As deviceEventClassId, ev.event_type_display_name As event_subcode, ev.descr
As msg,

CASE

      WHEN ev.rise_time is not NULL THEN
DATEADD(hour,DATEDIFF(hour,GETUTCDATE(),GETDATE()),ev.rise_time )

         ELSE ev.rise_time

      END

   AS endTime,

   CASE

      WHEN ev.registration_time is not NULL

         THEN DATEADD(hour,DATEDIFF(hour,GETUTCDATE(),GETDATE()),ev.registration_time )

         ELSE ev.registration_time

      END

   AS kscRegistrationTime,

   cast(ev.par7 as varchar(4000)) as sourceUserName,

   hs.wstrWinName as dHost,

   hs.wstrWinDomain as strNtDom, serv.wstrWinName As kscName,

      CAST(hs.nlp / 256 / 256 / 256 % 256 AS VARCHAR) + '.' +

   CAST(hs.nlp / 256 / 256 % 256 AS VARCHAR) + '.' +

   CAST(hs.nlp / 256 % 256 AS VARCHAR) + '.' +
```

```
    CAST(hs.nlp % 256 AS VARCHAR) AS sourceAddress,

    serv.wstrWinDomain as kscNtDomain,

      CAST(serv.nlp / 256 / 256 / 256 % 256 AS VARCHAR) + '.' +

    CAST(serv.nlp / 256 / 256 % 256 AS VARCHAR) + '.' +

    CAST(serv.nlp / 256 % 256 AS VARCHAR) + '.' +

    CAST(serv.nlp % 256 AS VARCHAR) AS kscIP,

    CASE

    WHEN virus.tmVirusFoundTime is not NULL

          THEN DATEADD(hour,DATEDIFF(hour,GETUTCDATE(),GETDATE()),virus.tmVirusFoundTime )

          ELSE ev.registration_time

      END

    AS virusTime,

    virus.wstrObject As filePath,

    virus.wstrVirusName as virusName,

    virus.result_ev as result

  FROM KAV.dbo.ev_event as ev

  LEFT JOIN KAV.dbo.v_akpub_host as hs ON ev.nHostId = hs.nId

  INNER JOIN KAV.dbo.v_akpub_host As serv ON serv.nId = 1

  Left Join KAV.dbo.rpt_viract_index as Virus on ev.event_id = virus.nEventVirus

  where registration_time >= DATEADD(minute, -191, GetDate())
```

c. Place the cursor in the **URL** field and in the displayed list, click in the line of the secret that you are using.

d. This opens the **Secret** window; in that window, in the **URL** field, specify the server connection address in the following format:

   `sqlserver://user:password@kscdb.example.com:1433/database`

   where:

   - `user`—user account with public and db_datareader rights to the required database.
   - `password`—user account password.
   - `kscdb.example.com:1433`—address and port of the database server.
   - `database`—name of the Kaspersky Security Center database. 'KAV' by default.

Click **Save**.

e. In the **Create connector** window, in the **Connection** section, in the **Query** field, replace the 'KAV' database name with the name of the Kaspersky Security Center database you are using.

You must do this if you want to use the ID column to which the query refers.

Click **Save**.

2. Install the collector in the web interface:

a. Start the Collector Installation Wizard in one of the following ways:

- In OSMP Console, in the **Monitoring** → **Resources** section, click **Add event source**.
- In OSMP Console, in the **Monitoring** → **Resources** → **Collectors** section, click **Add collector**.

b. At step 1 of the Installation Wizard, **Connect event sources**, specify the collector name and select the tenant.

c. At step 2 of the Installation Wizard, **Transport**, select the copy of the connector that you created at step 1.

d. At step 3 of the Installation Wizard, **Event parsing**, on the **Parsing schemes** tab, click **Add event parsing**.

e. This opens the **Basic event parsing** window; in that window, on the **Normalization scheme** tab, select **[OOTB] KSC from SQL** in the **Normalizer** drop-down list and click **OK**.

f. If necessary, specify the other settings in accordance with your requirements for the collector. For the purpose of importing events, editing settings at the remaining steps of the Installation Wizard is optional.

g. At step 8 of the Installation Wizard, **Setup validation**, click **Create and save service**.

The lower part of the window displays the command that you must use to install the collector on the server. Copy this command to the clipboard.

h. Close the Collector Installation Wizard by clicking **Save collector**.

3. Install the collector on the server.

To do so, on the server on which you want to receive Kaspersky Security Center events, run the command that you copied to the clipboard after creating the collector in the web interface.

As a result, the collector is installed and can receive events from the SQL database of Kaspersky Security Center.

You can view Kaspersky Security Center events in the **Events** section of the web interface.

Kaspersky Endpoint Detection and Response integration

Kaspersky Endpoint Detection and Response (hereinafter also referred to as "KEDR") is a functional unit of Kaspersky Anti Targeted Attack Platform that protects assets in an enterprise LAN.

You can configure KUMA integration with Kaspersky Endpoint Detection and Response to manage threat response actions on assets connected to Kaspersky Endpoint Detection and Response servers, and on Kaspersky Security Center assets. Commands to perform operations are received by the Kaspersky Endpoint Detection and Response server, which then relays those commands to the Kaspersky Endpoint Agent installed on assets.

You can also import events to KUMA and receive information about Kaspersky Endpoint Detection and Response alerts (for more details about alerts, see the *Configuring integration with an SIEM system* section of the Kaspersky Anti Targeted Attack Platform Help).

When KUMA is integrated with Kaspersky Endpoint Detection and Response, you can perform the following operations on Kaspersky Endpoint Detection and Response assets that have Kaspersky Endpoint Agent:

- Manage network isolation of assets.
- Manage prevention rules.
- Start applications.

To get instructions on configuring integration for response action management, contact your account manager or Technical Support.

Importing Kaspersky Endpoint Detection and Response events using the kafka connector

When importing events from Kaspersky Endpoint Detection and Response, telemetry is transmitted in clear text and may be intercepted by an intruder.

Kaspersky Endpoint Detection and Response 4.0, 4.1, 5.0, and 5.1 events can be imported to KUMA using a Kafka connector.

Several limitations are applicable to the import of events from Kaspersky Endpoint Detection and Response 4.0 and 4.1:

- Import of events is available if the KATA and KEDR license keys are used in Kaspersky Endpoint Detection and Response.
- Import of events is **not** available if the Sensor component installed on a separate server is used as part of Kaspersky Endpoint Detection and Response.

To import events, perform the actions in Kaspersky Endpoint Detection and Response and in KUMA.

Importing events from Kaspersky Endpoint Detection and Response 4.0 or 4.1

To import Kaspersky Endpoint Detection and Response 4.0 or 4.1 events to KUMA:

In Kaspersky Endpoint Detection and Response:

1. Use SSH or a terminal to log in to the management console of the Central Node server from which you want to export events.
2. When prompted by the system, enter the administrator account name and the password that was set during installation of Kaspersky Endpoint Detection and Response.

   The program component administrator menu is displayed.
3. In the program component administrator menu, select **Technical Support Mode**.

4. Press **Enter**.

   The Technical Support Mode confirmation window opens.

5. Confirm that you want to operate the application in Technical Support Mode. To do so, select **Yes** and press **Enter**.

6. Run the following command:

   ```
   sudo -i
   ```

7. In the `/etc/sysconfig/apt-services` configuration file, in the `KAFKA_PORTS` field, delete the value `10000`.

   If Secondary Central Node servers or the Sensor component installed on a separate server are connected to the Central Node server, you need to allow the connection with the server where you modified the configuration file via port 10000.

   We do not recommend using this port for any external connections other than KUMA. To restrict connections over port 10000 only for KUMA, run the following command:

   ```
   iptables -I INPUT -p tcp ! -s KUMA_IP_address --dport 10000 -j DROP
   ```

8. In the configuration file `/usr/bin/apt-start-sedr-iptables` add the value `10000` in the `WEB_PORTS` field, separated by a comma without a space.

9. Run the following command:

   ```
   sudo sh /usr/bin/apt-start-sedr-iptables
   ```

Preparations for exporting events on the Kaspersky Endpoint Detection and Response side are now complete.

In the SIEM component of Kaspersky Next XDR Expert:

1. On the KUMA Core server, add the IP address of the Central Node server in the format `<IP address> centralnode` to one of the following files:

   - `%WINDIR%\System32\drivers\etc\hosts`—for Windows.
   - `/etc/hosts file`—for Linux.

2. In OSMP Console, create a connector of the Kafka type.

   When creating a connector, specify the following parameters:

   - In the **URL** field, specify `<Central Node server IP address>:10000`.
   - In the **Topic** field, specify `EndpointEnrichedEventsTopic`.
   - In the **Consumer group** field, specify any unique name.

3. In OSMP Console, create a collector.

   Use the connector created at the previous step as the transport for the collector. Use [OOTB] KEDR telemetry as the normalizer for the collector.

If the collector is successfully created and installed, Kaspersky Endpoint Detection and Response events will be imported into KUMA. You can find and view these events in the events table.

Importing events from Kaspersky Endpoint Detection and Response 5.0 and 5.1

Several limitations apply when importing events from Kaspersky Endpoint Detection and Response 5.0 and 5.1:

- Import of events is available only for the non-high-availability version of Kaspersky Endpoint Detection and Response.
- Import of events is available if the KATA and KEDR license keys are used in Kaspersky Endpoint Detection and Response.
- Import of events is **not** available if the Sensor component installed on a separate server is used as part of Kaspersky Endpoint Detection and Response.

To import Kaspersky Endpoint Detection and Response 5.0 or 5.1 events to KUMA:

In Kaspersky Endpoint Detection and Response:

1. Use SSH or a terminal to log in to the management console of the Central Node server from which you want to export events.
2. When prompted by the system, enter the administrator account name and the password that was set during installation of Kaspersky Endpoint Detection and Response.

   The program component administrator menu is displayed.
3. In the program component administrator menu, select **Technical Support Mode**.
4. Press **Enter**.

   The Technical Support Mode confirmation window opens.
5. Confirm that you want to operate the application in Technical Support Mode. To do so, select **Yes** and press **Enter**.
6. In the `/usr/local/lib/python3.8/dist-packages/firewall/create_iptables_rules.py` configuration file, specify the additional port `10000` for the `WEB_PORTS` constant:

   ```
   WEB_PORTS = f'10000,80,{AppPort.APT_AGENT_PORT},{AppPort.APT_GUI_PORT}'
   ```

   `You do not need to perform this step for Kaspersky Endpoint Detection and Response 5.1 because the port is specified by default.`
7. Run the following commands:

   ```
   kata-firewall stop
   kata-firewall start --cluster-subnet <network mask for addressing cluster servers>
   ```

Preparations for exporting events on the Kaspersky Endpoint Detection and Response side are now complete.

In the SIEM component of Kaspersky Next XDR Expert:

1. On the KUMA Core server, add the IP address of the Central Node server in the format `<IP address> kafka.services.external.dyn.kata` to one of the following files:

   - `%WINDIR%\System32\drivers\etc\hosts`—for Windows.
   - `/etc/hosts file`—for Linux.

2. In OSMP Console, create a connector of the Kafka type.

   When creating a connector, specify the following parameters:

   - In the **URL** field, specify `<Central Node server IP address>:10000`.
   - In the **Topic** field, specify `EndpointEnrichedEventsTopic`.
   - In the **Consumer group** field, specify any unique name.

3. In OSMP Console, create a collector.

   Use the connector created at the previous step as the transport for the collector. It is recommended to use the *[OOTB] KEDR telemetry* normalizer as the normalizer for the collector.

If the collector is successfully created and installed, Kaspersky Endpoint Detection and Response events will be imported into KUMA. You can find and view these events in the events table.

Importing Kaspersky Endpoint Detection and Response events using the kata/edr connector

To use the kata/edr connector to import events of Kaspersky Endpoint Detection and Response 5.1 or earlier from hosts:

1. Configure event receipt in the SIEM component. To do this, in OSMP Console, create and install a collector with the 'kata/edr' connector or edit an existing collector, then save the modified settings and restart the collector.

2. On the KEDR side, accept the authorization request from the SIEM component to begin receiving events in Kaspersky Next XDR Expert.

As a result, the integration is configured and KEDR events start arriving in Kaspersky Next XDR Expert.

Creating a collector for receiving events from KEDR

To create a collector for receiving events from KEDR:

1. In OSMP Console, go to **Resources and services** → **Collectors**, select **Add collector**.

2. This opens the **Create collector** window; in that window, at step 1 **Connect event sources**, specify an arbitrary **Collector name** and in the drop-down list, select the appropriate **Tenant**.

3. At step 2 **Transport**, do the following:

   1. On the **Basic settings** tab:

      a. In the **Connector** field, select **Create** or start typing the name of the connector if you want to use a previously created connector.

      b. In the **Connector type** drop-down list, select the **kata/edr** connector. After you select the kata/edr connector type, more fields to fill in are displayed.

      c. In the **URL** field, specify the address for connecting to the KEDR server in the following `<name or IP address of the host>:<connection port, 443 by default>` format. If the KEDR solution is deployed in a cluster, you can click **Add** to add all nodes. KUMA will connect to each specified node in sequence. If the KEDR solution is installed in a distributed configuration, on the KUMA side, you must configure a separate collector for each KEDR server.

      d. In the **Secret** field, select **Create** to create a new secret. This opens the **Create secret** window; in that window, specify the **Name** and click **Generate and download a certificate and private encryption key**.

         As a result, the certificate.zip archive is downloaded to the browser's Downloads folder; the archive contains the 'key.pem' key file and the 'cert.pem' certificate file. Extract the archive. Click **Upload certificate** and select the cert.pem file. Click **Upload private key** and select the key.pem file. Click **Create**; the secret is added to the **Secret** drop-down list and automatically selected.

         You can also select the created secret from the **Secret** list. KUMA uses the selected secret to connect to KEDR.

      e. The **External ID** field contains the ID for external systems. This ID is displayed in the KEDR web interface when authorizing the KUMA server. KUMA generates an ID automatically and the **External ID** field is automatically pre-populated.

   2. On the **Advanced settings** tab:

      a. To get detailed information in the collector log, move the **Debug** toggle switch to the enabled position.

      b. If necessary, in the **Character encoding** field, select the encoding of the source data to be converted to UTF-8. We only recommend configuring a conversion if you find invalid characters in the fields of the normalized event. By default, no value is selected.

      c. Specify the maximum **Number of events** per one request to KEDR. The default value is 0, which means that KUMA uses the value specified on the KEDR server. For details, please refer to KATA Help. You can specify an arbitrary value that must not exceed the value on the KEDR side. If the value you specify exceeds the value of the **Maximum number of events** setting specified on the KEDR server, the KUMA collector log will display the error *Bad Request: max_events N is greater than the allowed value*.

      d. Fill in the **Events fetch timeout** field to receive events after a specified period of time. The default value is 0. This means that the default value of the KEDR server is applied. For details, please refer to KATA Help. This field specifies the time after which the KEDR server must send events to KUMA. The KEDR server uses two parameters: the maximum number of events and the events fetch timeout. Events are sent when the specified number of events is collected or the configured time elapses, whichever happens first. If the specified time has elapsed, but the specified number of events has not been collected, the KEDR server sends the events that it already has, without waiting for more.

e. In the **Client timeout** field, specify how long KUMA must wait for a response from the KEDR server, in seconds. Default value: 1,800 s; displayed as 0. The client-side limit is specified in the **Client timeout** field. The **Client timeout** must be greater than the **Events fetch timeout** of the server to wait for the server's response without interrupting the current event collection task with a new request. If the response from the KEDR server does not arrive in the end, KUMA repeats the request.

f. In the **KEDRQL filter** field, specify the conditions for filtering the request. As a result, pre-filtered events are received from KEDR. For details about available filter fields, please refer to the KATA Help.

4. At step 3 **Parsing**, click **Add event parsing** and select **[OOTB] KEDR telemetry** in the **Basic event parsing** window.

5. To finish creating the collector in the web interface, click **Create and save service**. Then copy the collector installation command from the web interface and run this installation command on the command line on the server where you want to install the collector.

   If you were editing an existing collector, click **Save and restart services**.

As a result, the collector is created and is ready to send requests; the collector is displayed in the **Resources and services** → **Active services** section with the yellow status until KEDR accepts an authorization request from KUMA.

Authorizing KUMA on the KEDR side

After the collector is created in KUMA, for requests from KUMA to start arriving to KEDR, the KUMA authorization request must be accepted on the KEDR side. With the authorization request accepted, the KUMA collector automatically sends scheduled requests to KEDR and waits for a response. While waiting, the status of the collector is yellow, and after receiving the first response to a request, the status of the collector turns green.

As a result, the integration is configured and you can view events arriving from KEDR in the **Events** section of OSMP Console.

The initial request fetches part of the historical events that had occurred before the integration was configured. Current events begin arriving after all of the historical events. If you change the value of the **URL** setting or the **External ID** of an existing collector, KEDR treats the next request as an initial request, and after starting the KUMA collector with the modified settings, you will receive part of the historical events all over again. If you do not want to receive historical events, go to the settings of the relevant collector, configure the mapping of the KEDR and KUMA `timestamp` fields in the normalizer, and specify a filter by `timestamp` at the 'Event filtering' step of the collector Installation Wizard — the `timestamp` of the event must be greater than the `timestamp` when the collector is started.

Possible errors and solutions

If in the collector log, you see the *Conflict: An external system with the following ip and certificate digest already exists. Either delete it or provide a new certificate* error, create a new secret with the a certificate in the connector of the collector.

If in the collector log, you see the *Continuation token not found* error in response to an event request, create a new connector, attach it to the collector and restart the collector; alternatively, create a new secret with a new certificate in the connector of the collector. If you do not want to receive events generated before the error occurred, configure a Timestamp filter in the collector.

Configuring the display of a link to a Kaspersky Endpoint Detection and Response detection in the KUMA alert

When Kaspersky Endpoint Detection and Response detections are received, KUMA creates an alert for each detection. You can configure the display of a link to a Kaspersky Endpoint Detection and Response detection in KUMA alert information.

> You can configure the display of a detection link if you use only one Central Node server in Kaspersky Endpoint Detection and Response. If Kaspersky Endpoint Detection and Response is used in a distributed solution mode, it is impossible to configure the display of the links to Kaspersky Endpoint Detection and Response detections in KUMA.

To configure the display of a link to a detection in KUMA alert details, you need to complete steps in the Kaspersky Endpoint Detection and Response web interface and KUMA.

In the Kaspersky Endpoint Detection and Response web interface, you need to configure the integration of the application with KUMA as a SIEM system. For details on configuring integration, refer to the *Kaspersky Anti Targeted Attack Platform* documentation, *Configuring integration with a SIEM system* section.

Configuring the display of a link in the OSMP Console includes the following steps:

1. Adding an asset that contains information about the Kaspersky Endpoint Detection and Response Central Node server from which you want to receive detections, and assigning a category to that asset.

2. Creating a correlation rule.

3. Creating a correlator.

You can use a pre-configured correlation rule. In this case configuring the display of a link in the OSMP Console includes the following steps:

1. Creating a correlator.

   Select the `[OOTB] KATA Alert` correlation rule.

2. Adding an asset that contains information about the Kaspersky Endpoint Detection and Response Central Node server from which you want to receive detections and assigning a category `KATA standAlone` to that asset.

Step 1. Adding an asset and assigning a category to it

First, you need to create a category that will be assigned to the asset being added. Then add an asset.

To add an asset:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Detection and response**.

2. Click the **Add** button.

   The **Add asset** details area opens in the right part of the window.

3. Define the following asset parameters:

   a. In the **Asset name** field, enter an asset name.

   b. In the **Tenant** drop-down list, select the tenant that will own the asset.

   c. In the **IP address** field, specify the IP address of the Kaspersky Endpoint Detection and Response Central Node server from which you want to receive detections.

   d. In the **Categories** field, select the category that you added in the previous step.

   > If you are using a predefined correlation rule, you need to select the `KATA standAlone` category.

   e. If required, define the values for the following fields:

   - In the **FQDN** field, specify the Fully Qualified Domain Name of the Kaspersky Endpoint Detection and Response server.
   - In the **MAC address** field, specify the MAC address of the Central Node Kaspersky Endpoint Detection and Response Central Node server.
   - In the **Owner** field, define the name of the asset owner.

4. Click the **Save** button.

Step 2. Adding a correlation rule

To add a correlation rule:

1. In the main menu, go to **Monitoring** → **Resources and services**.
2. Select **Correlation rules** and click the **Create correlation rule** button.
3. On the **General** tab, specify the following settings:

   a. In the **Name** field, define the rule name.

   b. In the **Type** drop-down list, select **simple**.

   c. In the **Propagated fields** field, add the following fields: DeviceProduct, DeviceAddress, EventOutcome, SourceAssetID, DeviceAssetID.

   d. If required, define the values for the following fields:

   - In the **Rate limit** field, define the maximum number of times per second that the rule will be triggered.
   - In the **Severity** field, define the severity of alerts and correlation events that will be created as a result of the rule being triggered.
   - In the **Description** field, provide any additional information.

4. On the **Selectors** → **Settings** tab, specify the following settings:

   a. In the **Filter** drop-down list, select **Create new**.

   b. In the **Conditions** field, click the **Add group** button.

   c. In the operator field for the group you added, select **AND**.

   d. Add a condition for filtering by KATA value:

      1. In the **Conditions** field, click the **Add condition** button.

      2. In the condition field, select **If**.

      3. In the **Left operand** field, select **Event field**.

      4. In the **Event field** field, select **DeviceProduct**.

      5. In the **operator** field, select =.

      6. In the **Right operand** field, select **constant**.

      7. In the **value** field, enter KATA.

   e. Add a category filter condition:

      1. In the **Conditions** field, click the **Add condition** button.

      2. In the condition field, select **If**.

      3. In the **Left operand** field, select **Event field**.

      4. In the **Event field** field, select **DeviceAssetID**.

      5. In the **operator** field, select **inCategory**.

      6. In the **Right operand** field, select **constant**.

      7. Click the button.

      8. Select the category in which you placed the Kaspersky Endpoint Detection and Response Central Node server asset.

      9. Click the **Save** button.

   f. In the **Conditions** field, click the **Add group** button.

   g. In the operator field for the group you added, select **OR**.

   h. Add a condition for filtering by event class identifier:

      1. In the **Conditions** field, click the **Add condition** button.

      2. In the condition field, select **If**.

3. In the **Left operand** field, select **Event field**.

4. In the **Event field** field, select **DeviceEventClassID**.

5. In the **operator** field, select =.

6. In the **Right operand** field, select **constant**.

7. In the **value** field, enter **taaScanning**.

i. Repeat steps 1–7 in F for each of the following event class IDs:

- file_web.
- file_mail.
- file_endpoint.
- file_external.
- ids.
- url_web.
- url_mail.
- dns.
- iocScanningEP.
- yaraScanningEP.

5. On the **Actions** tab, specify the following settings:

a. In the **Actions** section, open the **On every event** drop-down list.

b. Select the **Output** check box.

c. In the **Enrichment** section, click the **Add enrichment** button.

d. In the **Source kind** drop-down list, select **template**.

e. In the **Template** field, enter *https://{{.DeviceAddress}}:8443/katap/#/alerts?id={{.EventOutcome}}*.

f. In the **Target field** drop-down list, select **DeviceExternalID**.

g. If necessary, turn on the **Debug** toggle switch to log information related to the operation of the resource.

6. Click the **Save** button.

Step 3. Creating a correlator

You need to launch the correlator Installation Wizard. At step 3 of the wizard, you are required to select the correlation rule that you added by following this guide.

After the correlator is created, a link to these detections will be displayed in the details of alerts created when receiving detections from Kaspersky Endpoint Detection and Response. The link is displayed in the correlation event details (**Threat hunting** section), in the **DeviceExternalID** field.

> If you want the FQDN of the Kaspersky Endpoint Detection and Response Central Node server to be displayed in the DeviceHostName field, in the detection details, you need to create a DNS record for the server and create a DNS enrichment rule at step 4 of the wizard.

Integration with Kaspersky CyberTrace

Kaspersky CyberTrace (hereinafter CyberTrace) is a tool that integrates threat data streams with SIEM solutions. It provides users with instant access to analytics data, increasing their awareness of security decisions.

You can integrate CyberTrace with KUMA in one of the following ways:

- Integrate CyberTrace indicator search feature to enrich KUMA events with information from CyberTrace data streams.
- Integrate the entire CyberTrace web interface into KUMA to get full access to CyberTrace.

  > CyberTrace console integration is available only if your CyberTrace license includes multi-user feature.

Integrating CyberTrace indicator search

To integrate CyberTrace indicator search:

1. Configure CyberTrace to receive and process KUMA requests.

   You can configure the integration with KUMA immediately after installing CyberTrace in the Quick Start Wizard or later in the CyberTrace web interface.

2. Create an event enrichment rule.

   In the enrichment rule, you can specify which data from CyberTrace you want to enrich the event with.

3. Create a collector to receive events that you want to enrich with CyberTrace data.

4. Link the enrichment rule to the collector.

5. Save and create the service:

   - If you linked the rule to a new collector, click **Save and create**, copy the collector ID in the opened window and use the copied ID to install the collector on the server using the command line interface.
   - If you linked the rule to an existing collector, click **Save and restart services** to apply the settings.

   The configuration of the integration of CyberTrace indicator search is complete and KUMA events will be enriched with CyberTrace data.

Example of testing CyberTrace data enrichment.

By default, KUMA does not test the connection with CyberTrace.

If you want to test the integration with CyberTrace and make sure that event enrichment is working, you can follow the steps of the following example or adapt the example to your situation. The example shows an integration test, which performs enrichment and shows that the event contains the specified test URL.

To run the test:

1. Create a test enrichment rule with parameters listed in the table below.

Setting	Value
Name	Test CT enrichment
Tenant	Shared
Source kind	cybertrace-http
URL	<URL of the cybertrace server to which you want to send requests>:9999
Mapping	KUMA field: RequestURL CyberTrace indicator: url
Debug	Enabled

1. Create a test collector with the following parameters:

   At step **2 Transport**, specify the http connector.

   At step **3 Parsing**, specify the normalizer and select the json parsing method, set the mapping of the RequestUrl – RequestUrl fields.

   At step **6 Enrichment**, specify the 'Test CT enrichment' rule.

   At step **7 Routing**, specify the storage where events must be sent.

2. Click **Create and save service**.

   A complete command for installing the collector is displayed in the window.

3. Click **Copy** to copy the command to the clipboard and run the command on the command line. Wait for the command to complete, return to the KUMA Console, and click **Save collector**.

   A test collector is created and the test enrichment rule is linked to the collector.

4. Use the command line interface to send a request to the collector, which will trigger an event, which will then be enriched with the test URL `http://fakess123bn.nu`. For example:

   ```
   curl --request POST \
     --url http://<ID of the host where the collector is installed>:<port of the collector>/input \
     --header 'Content-Type: application/json' \
     --data '{"RequestUrl":"http://fakess123bn.nu"}'
   ```

5. Go to the KUMA **Events** section and run the following query to filter event output and find the enriched event:

   ```
   SELECT * FROM `events` WHERE RequestUrl = 'http://fakess123bn.nu' ORDER BY Timestamp DESC LIMIT 250
   ```

   Result:

   Enrichment is successful, the event now has a **RequestURL** field with the `http://fakess123bn.nu` value, as well as a TI indicator and indicator category with CyberTrace data.

If the test did not result in enrichment, for example, if the TI indicator is missing, we recommend to do the following:

1. Check the settings of the collector and enrichment rules.
2. Download the collector logs using the following command and look for errors in the logs:

   ```
   tail -f /opt/kaspersky/kuma/collector/<collector ID>/log/collector
   ```

Configuring CyberTrace to receive and process requests

You can configure CyberTrace to receive and process requests from KUMA immediately after its installation in the Quick Start Wizard or later in the program web interface.

To configure CyberTrace to receive and process requests in the Quick Start Wizard:

1. Wait for the CyberTrace Quick Start Wizard to start after the program is installed.

   The **Welcome to Kaspersky CyberTrace** window opens.

2. In the **select SIEM** drop-down list, select the type of SIEM system from which you want to receive data and click the **Next** button.

   The **Connection Settings** window opens.

3. Do the following:

   a. In the **Service listens on** settings block, select the **IP and port** option.

   b. In the **IP address** field, enter `0.0.0.0`.

   c. In the **Port** field, enter the port for receiving events, the default port is `9999`.

   d. Under **Service sends events to**, specify `127.0.0.1` in the **IP address or hostname** field and in the **Port** field, specify `9998`.

   Leave the default values for everything else.

   e. Click **Next**.

   The **Proxy Settings** window opens.

4. If a proxy server is being used in your organization, define the settings for connecting to it. If not, leave all the fields blank and click **Next**.

   The **Licensing Settings** window opens.

5. In the **Kaspersky CyberTrace license key** field, add a license key for CyberTrace.

6. In the **Kaspersky Threat Data Feeds certificate** field, add a certificate that allows you to download updated data feeds from servers, and click **Next**.

CyberTrace will be configured.

To configure CyberTrace to receive and process requests in the program web interface:

1. In the CyberTrace web interface, select **Settings** – **Service**.
2. In the **Connection Settings** block:
   a. Select the **IP and port** option.
   b. In the **IP address** field, enter `0.0.0.0`.
   c. In the **Port** field, specify the port for receiving events, the default port is `9999`.
3. In the **Web interface** settings block, in the **IP address or hostname** field, enter `127.0.0.1`.
4. In the upper toolbar, click **Restart the CyberTrace Service**.
5. Select **Settings** – **Events format**.
6. In the **Alert events format** field, enter `%Date% alert=%Alert%%RecordContext%`.
7. In the **Detection events format** field, enter `Category=%Category%|MatchedIndicator=%MatchedIndicator%%RecordContext%`.
8. In the **Records context format** field, enter `|%ParamName%=%ParamValue%`.
9. In the **Actionable fields context format** field, enter `%ParamName%:%ParamValue%`.

CyberTrace will be configured.

After updating CyberTrace configuration you have to restart the CyberTrace server.

Creating event Enrichment rules

To create event enrichment rules:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Enrichment rules**.
2. In the left part of the window, select or create a folder for the new rule.
   The list of available enrichment rules will be displayed.
3. Click **Create** to create a new rule.
   The enrichment rule window will be displayed.
4. Enter the rule configuration parameters:
   a. In the **Name** field, enter a unique name for the rule. The name must contain 1 to 128 Unicode characters.
   b. In the **Tenant** drop-down list, select the tenant that will own this resource.
   c. In the **Source kind** drop-down list, select **cybertrace**.

d. Specify the **URL** of the CyberTrace server to which you want to connect. For example, *example.domain.com:9999*.

e. If necessary, use the **Number of connections** field to specify the maximum number of connections to the CyberTrace server that can be simultaneously established by OSMP. By default, this value is equal to the number of vCPUs of the KUMA Core server.

f. In the **RPS** field, enter the number of requests to the CyberTrace server per second that OSMP can make. The default value is `1000`.

g. In the **Maximum number of events in enrichment queue** field, specify the maximum number of events stored in the enrichment queue for re-sending. The default setting is `1000000`.

h. In the **Timeout** field, specify the maximum number of seconds OSMP should wait for a response from the CyberTrace server. Until a response is received or the time expires, the event is not sent to the Correlator. If a response is received before the timeout, it is added to the `TI` field of the event and the event processing continues. The default value is `30`.

i. Use the **Debug** drop-down list to indicate whether or not to enable logging of service operations. Logging is disabled by default.

j. In the **Tags** drop-down list, select the tags for the event enrichment rule.

k. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

l. In the **Mapping** settings block, you must specify the fields of events to be checked via CyberTrace, and define the rules for mapping fields of KUMA events to CyberTrace indicator types:

- In the **KUMA field** column, select the field whose value must be sent to CyberTrace.
- In the **CyberTrace indicator** column, select the CyberTrace indicator type for every field you selected:
  - **ip**
  - **url**
  - **hash**

You must provide at least one string to the table. You can click the **Add** button to add a string, and can click the × button to remove a string.

m. In the **Filter** section, you can specify conditions to identify events that will be processed using the enrichment rule. You can select an existing filter from the **Filter** drop-down list or **Create** a new filter.

Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.
2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.
3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select an operator.

   Filter operators

   - **=**—the left operand equals the right operand.
   - **<**—the left operand is less than the right operand.
   - **<=**—the left operand is less than or equal to the right operand.
   - **>**—the left operand is greater than the right operand.
   - **>=**—the left operand is greater than or equal to the right operand.
   - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
   - **contains**—the left operand contains values of the right operand.
   - **startsWith**—the left operand starts with one of the values of the right operand.
   - **endsWith**—the left operand ends with one of the values of the right operand.
   - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
   - **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

     The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

     If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
   - **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

     If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
   - **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
   - **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

5. Click **Create new**.

A new enrichment rule will be created.

CyberTrace indicator search integration is now configured. You can now add the created enrichment rule to a collector. You must restart KUMA collectors to apply the new settings.

If any of the CyberTrace fields in the events details area contains "[{" or "}]" values, it means that information from CyberTrace data feed was processed incorrectly and it's possible that some of the data is not displayed. You can get all data feed information by copying the events **TI indicator** field value from KUMA and searching for it in the CyberTrace in the indicators section. All relevant information will be displayed in the **Indicator context** section of CyberTrace.

Integrating CyberTrace interface

You can integrate the CyberTrace web interface with OSMP Console. When this integration is enabled, OSMP Console includes a **CyberTrace** section that provides access to the CyberTrace web interface. You can configure the integration in the **Settings** → **KUMA** → **Open KUMA settings** → **Kaspersky CyberTrace** section of OSMP Console.

To integrate the CyberTrace web interface in OSMP Console:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Secrets**.

   The list of available secrets will be displayed.

2. Click the **Create** button to create a new secret. This resource is used to store credentials of the CyberTrace server.

   The secret window is displayed.

3. Enter information about the secret:

   a. In the **Name** field, choose a name for the added secret. The name must contain 1 to 128 Unicode characters.

   b. In the **Tenant** drop-down list, select the tenant that will own this resource.

   c. In the **Kind** drop-down list, select **credentials**.

   d. In the **User** and **Password** fields, enter credentials for your CyberTrace server.

   e. If necessary, select the tags for the secret from the **Tags** drop-down list

   f. If necessary, in the **Description** field, add up to 4,000 Unicode characters describing the resource.

4. Click **Create new**.

   The CyberTrace server credentials are now saved and can be used in other KUMA resources.

5. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

6. Under **Threat intelligence**, click **Kaspersky CyberTrace**.

   This opens the **Kaspersky CyberTrace integration** window.

7. Make the necessary changes to the following parameters:

   - **State**—enable the toggle switch if you want to integrate the CyberTrace web interface into OSMP Console.
   - **Host** (required)—enter the address of the CyberTrace server.
   - **Port** (required)—enter the port of the CyberTrace server; the default port for managing the web interface is 443.
   - In the **Secret** drop-down list, select the secret you created before.
   - In the **Allow hosts** drop-down list, select the hosts that support the CORS technology.
   - In the **Major version of CyberTrace** field, select the CyberTrace version. Possible values: **<5**, **≥ 5.0**. If version **≥ 5.0** is selected, requests are executed in the General tenant in CyberTrace and use the new version of the API provided by Kaspersky CyberTrace 5.0 and later. If you select **<5**, requests use the Kaspersky CyberTrace 4.x API. The default value is **<5**.

8. You can configure access to the CyberTrace web interface in the following ways:

   - Use hostname or IP when logging into OSMP Console.

     To do this, in the **Allow hosts** section, click **Add host** and in the field that is displayed, enter the IP or hostname of the device.

   - Use FQDN when logging into OSMP Console.

     If you are using the Mozilla Firefox browser to manage the console, the CyberTrace section may fail to display data. In this case, configure the data display (see below).

9. Click **Save**.

CyberTrace is now integrated with OSMP, and the **CyberTrace** section is displayed in OSMP Console.

*To configure the data display in the **CyberTrace** section when using the FQDN to log in to OSMP Console in Mozilla Firefox:*

1. Clear your browser cache.

2. In the browser's address bar, enter the FQDN of OSMP Console with port number.

   A window will open to warn you of a potential security threat.

3. Click the **Details** button.

4. In the lower part of the window, click the **Accept risk and continue** button.

   An exclusion is created for the URL of OSMP Console.

5. In the browser's address bar, enter the URL of OSMP Console with port number.

6. Go to the **CyberTrace** section.

Data will be displayed in this section.

Updating CyberTrace deny list (Internal TI)

When the CyberTrace web interface is integrated into OSMP Console, you can update the CyberTrace denylist or **Internal TI** with information from KUMA events.

To update CyberTrace Internal TI:

1. Open the event details area from the events table, Alert window, or correlation event window and click the link on a domain, web address, IP address, or file hash.

   The context menu opens.

2. Select **Add to Internal TI of CyberTrace**.

The selected object is now added to the CyberTrace denylist.

Integration with Kaspersky Threat Intelligence Portal

The Kaspersky Threat Intelligence Portal combines all of Kaspersky's knowledge about cyberthreats and how they're related into a single web service. When integrated with KUMA, it helps users to make faster and better-informed decisions, providing them with data about URLs, domains, IP addresses, WHOIS / DNS data.

> Access to the Kaspersky Threat Intelligence Portal is provided based on a fee. License certificates are created by Kaspersky experts. To obtain a certificate for Kaspersky Threat Intelligence Portal, contact your Technical Account Manager.

Initializing integration

To integrate Kaspersky Threat Intelligence Portal into KUMA:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Secrets**.

   The list of available secrets will be displayed.

2. Click the **Add secret** button to create a new secret. This resource is used to store credentials of your Kaspersky Threat Intelligence Portal account.

   The secret window is displayed.

3. Enter information about the secret:

   a. In the **Name** field, choose a name for the added secret.

   b. In the **Tenant** drop-down list, select the tenant that will own the created resource.

   c. In the **Type** drop-down list, select **ktl**.

   d. In the **User** and **Password** fields, enter credentials for your Kaspersky Threat Intelligence Portal account.

   e. If you want, enter a **Description** of the secret.

4. Upload your Kaspersky Threat Intelligence Portal certificate key:

   a. Click the **Upload PFX** button and select the PFX file with your certificate.

      The name of the selected file appears to the right of the **Upload PFX** button.

   b. Enter the password to the PFX file in the **PFX password** field.

5. Click **Save**.

   The Kaspersky Threat Intelligence Portal account credentials are now saved and can be used in other KUMA resources.

6. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

7. Under **Threat intelligence**, click **Kaspersky Threat Lookup**.

   This opens the **Kaspersky Threat Lookup integration** window.

8. Make sure the status is enabled.

9. In the **Secret** drop-down list, select the secret you created before.

   You can create a new secret by clicking the button with the plus sign. The created secret will be saved in the **Monitoring** → **Resources and services** → **Secrets** section.

10. If necessary, select a proxy server in the **Proxy** drop-down list.

11. Click **Save**.

12. After you save the settings, log in to the console and accept the **Terms of Use**. Otherwise, an error is returned in the API.

The integration process of Kaspersky Threat Intelligence Portal with KUMA is completed.

Once Kaspersky Threat Intelligence Portal and KUMA are integrated, you can request additional information from the event details area about hosts, domains, URLs, IP addresses, and file hashes (MD5, SHA1, SHA256).

Requesting information from Kaspersky Threat Intelligence Portal

To request information from Kaspersky Threat Intelligence Portal:

1. Open the event details area from the events table, Alert window, or correlation event window and click the link on a domain, web address, IP address, or file hash.

   The **Threat Lookup enrichment** area opens in the right part of the screen.

2. Select check boxes next to the data types you want to request.

   If neither check box is selected, all information types are requested.

3. In the **Maximum number of records in each data group** field enter the number of entries per selected information type you want to receive. The default value is `10`.

4. Click **Request**.

A *kt1* task has been created. When it is completed, events are enriched with data from Kaspersky Threat Intelligence Portal which can be viewed from the events table, Alert window, or correlation event window.

Viewing information from Kaspersky Threat Intelligence Portal

To view information from Kaspersky Threat Intelligence Portal:

Open the event details area from the events table, alert window, or correlation event window and click the link on a domain, web address, IP address, or file hash for which you previously requested information from Kaspersky Threat Intelligence Portal.

The event details area opens in the right part of the screen with data from Kaspersky Threat Intelligence Portal; the time when it was received is indicated at the bottom of the screen.

Information received from Kaspersky Threat Intelligence Portal is cached. If you click a domain, web address, IP address, or file hash in the event details pane for which KUMA has information available, the data from Kaspersky Threat Intelligence Portal opens, with the time it was received indicated at the bottom, instead of the **Threat Lookup enrichment** window. You can update the data.

Updating information from Kaspersky Threat Intelligence Portal

To update information, received from Kaspersky Threat Intelligence Portal:

1. Open the event details area from the events table, alert window, or correlation event window and click the link on a domain, web address, IP address, or file hash for which you previously requested information from Kaspersky Threat Intelligence Portal.

2. Click **Update** in the event details area containing the data received from the Kaspersky Threat Intelligence Portal.

   The **Threat Lookup enrichment** area opens in the right part of the screen.

3. Select the check boxes next to the types of information you want to request.

   If neither check box is selected, all information types are requested.

4. In the **Maximum number of records in each data group** field enter the number of entries per selected information type you want to receive. The default value is `10`.

5. Click **Update**.

   The *KTL* task is created and the new data received from Kaspersky Threat Intelligence Portal is requested.

6. Close the **Threat Lookup enrichment** window and the details area with KTL information.

7. Open the event details area from the events table, Alert window or correlation event window and click the link on a domain, URL, IP address, or file hash for which you updated Kaspersky Threat Intelligence Portal information and select **Show info from Threat Lookup**.

The event details area opens on the right with data from Kaspersky Threat Intelligence Portal, indicating the time when it was received on the bottom of the screen.

Connecting over LDAP

LDAP connections are created and managed under **Settings** → **KUMA** → **Open KUMA settings** → **LDAP server** in OSMP Console. The **LDAP server integration by tenant** section shows the tenants for which LDAP connections were created. Tenants can be created or deleted.

If you select a tenant, the **LDAP server integration** window opens to show a table containing existing LDAP connections. Connections can be created or edited. In this window, you can change the frequency of queries sent to LDAP servers and set the retention period for obsolete data.

After integration is enabled, information about Active Directory accounts becomes available in the alert window, the correlation events detailed view window, and the incidents window. If you click an account name in the **Related users** section of the window, the **Account details** window opens with the data imported from Active Directory.

Data from LDAP can also be used when enriching events in collectors and in analytics.

Imported Active Directory attributes

The following account attributes can be requested from Active Directory:

- `accountExpires`
- `badPasswordTime`
- `cn`
- `company`
- `department`
- `displayName`
- `distinguishedName`
- `division`
- `employeeID`
- `ipaUniqueID`
- `l`
- `Mail`
- `mailNickname`
- `managedObjects`
- `manager`
- `memberOf` (this attribute can be used for search during correlation)
- `mobile`
- `objectGUID` (this attribute always requested from Active Directory even if a user doesn't specify it)
- `objectSID`
- `physicalDeliveryOfficeName`
- `sAMAccountName`
- `sAMAccountType`
- `sn`
- `streetAddress`

- `telephoneNumber`
- `title`
- `userAccountControl`
- `UserPrincipalName`
- `whenChanged`
- `whenCreated`

For details about Active Directory attributes, please refer to the Microsoft documentation.

Enabling and disabling LDAP integration

You can enable or disable all LDAP connections of the tenant at the same time, or enable and disable an LDAP connection individually.

To enable or disable all LDAP connections of a tenant:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**, and then select the tenant for which you want to enable or disable all LDAP connections.

   The **LDAP server integration** window opens.

3. Enable or disable the **State** toggle switch.

4. Click **Save**.

To enable or disable a specific LDAP connection:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**, and then select the tenant for which you want to enable or disable all LDAP connections.

   The **LDAP server integration** window opens.

3. In the connections table, enable or disable the **State** toggle switch for the relevant connection.

4. Click **Save**.

Adding a tenant to the LDAP server integration list

To add a tenant to the list of tenants for integration with an LDAP server:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.
   This opens the KUMA settings window.
2. Under **Integrations**, click **LDAP server**.
   The **LDAP server integration by tenant** window opens.
3. Click the **Add settings for a new tenant** button.
   The **LDAP server integration** window is displayed.
4. In the **Tenant** drop-down list, select the tenant that you need to add.
5. Click **Save**.

The selected tenant is added to the LDAP server integration list.

To delete a tenant from the list of tenants for integration with an LDAP server:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.
   This opens the KUMA settings window.
2. Under **Integrations**, click **LDAP server**.
   The **LDAP server integration by tenant** window opens.
3. Select the check box next to the tenant that you need to delete, and click **Delete**.
4. Confirm deletion of the tenant.

The selected tenant is deleted from the LDAP server integration list.

Creating an LDAP server connection

To create a new LDAP connection to Active Directory:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.
   This opens the KUMA settings window.
2. Under **Integrations**, click **LDAP server**.
   The **LDAP server integration by tenant** window opens.
3. Select or create a tenant for which you want to create a LDAP connection.
   The **LDAP server integration** window opens.
4. Click the **Add connection** button.
   The **Connection parameters** window opens.

5. Add a secret containing the account credentials for connecting to the Active Directory server. To do so:

   a. If you previously added a secret, in the **Secret** drop-down list, select the existing secret (with the **credentials** type).

      The selected secret can be changed by clicking the pencil icon ( ).

   b. If you want to create a new secret, click the plus icon (+).

      The **Secret** window opens.

   c. In the **Name** (required) field, enter the name of the secret containing 1 to 128 Unicode characters.

   d. In the **User** and **Password** (required) fields, enter the account credentials for connecting to the Active Directory server.

      You can enter the user name in one of the following formats: <user name>@<domain> or <domain><user name>.

   e. In the **Description** field, enter a description of up to 4,000 Unicode characters.

   f. Click the **Save** button.

6. In the **Name** (required) field, enter the unique name of the LDAP connection.

   The length of the string must be 1 to 128 Unicode characters.

7. In the **URL** (required) field, enter the address of the domain controller in the format `<hostname or IP address of server>:<port>`.

   > In case of server availability issues, you can specify multiple servers with domain controllers by separating them with commas. All of the specified servers must reside in the same domain.

8. If you want to use TLS encryption for the connection with the domain controller, select one of the following options from the **Type** drop-down list:

   - **startTLS**.

     When the startTLS method is used, first it establishes an unencrypted connection over port 389, then it sends an encryption request. If the STARTTLS command ends with an error, the connection is terminated.

     > Make sure that port 389 is open. Otherwise, a connection with the domain controller will be impossible.

   - **ssl**.

     When using SSL, an encrypted connection is immediately established over port 636.

   - **insecure**.

   When using an encrypted connection, it is impossible to specify an IP address as a URL.

9. If you enabled TLS encryption at the previous step, add a TLS certificate. To do so:

   a. If you previously uploaded a certificate, select it from the **Certificate** drop-down list.

      If no certificate was previously added, the drop-down list shows **No data**.

b. If you want to upload a new certificate, click the plus icon (+) on the right of the **Certificate** list.

The **Secret** window opens.

c. In the **Name** field, enter the name that will be displayed in the list of certificates after the certificate is added.

d. Click the **Upload certificate file** button to add the file containing the Active Directory certificate. X.509 certificate public keys in Base64 are supported.

e. If necessary, provide any relevant information about the certificate in the **Description** field.

f. Click the **Save** button.

The certificate will be uploaded and displayed in the **Certificate** list.

10. In the **Timeout in seconds** field, indicate the amount of time to wait for a response from the domain controller server.

    If multiple addresses are indicated in the **URL** field, KUMA will wait the specified number of seconds for a response from the first server. If no response is received during that time, the program will contact the next server. If none of the indicated servers responds during the specified amount of time, the connection will be terminated with an error.

11. In the **Base DN** field, enter the base distinguished name of the directory where the search request should be performed.

12. In the **Custom AD Account Attributes** field, specify the **additional attributes that you want to use to enrich events**.

    Before configuring event enrichment using custom attributes, make sure that custom attributes are configured in AD.

    To enrich events with accounts using custom attributes:

    1. Add **Custom AD Account Attributes** in the LDAP connection settings.

       Standard **imported attributes from AD** cannot be added as custom attributes. For example, if you add the standard `accountExpires` attribute as a custom attribute, KUMA returns an error when saving the connection settings.

       The following account attributes can be requested from Active Directory:

       - `accountExpires`
       - `badPasswordTime`
       - `cn`
       - `company`
       - `department`
       - `displayName`
       - `distinguishedName`
       - `division`

- `employeeID`
- `ipaUniqueID`
- `l`
- `Mail`
- `mailNickname`
- `managedObjects`
- `manager`
- `memberOf` (this attribute can be used for search during correlation)
- `mobile`
- `objectGUID` (this attribute always requested from Active Directory even if a user doesn't specify it)
- `objectSID`
- `physicalDeliveryOfficeName`
- `sAMAccountName`
- `sAMAccountType`
- `sn`
- `streetAddress`
- `telephoneNumber`
- `title`
- `userAccountControl`
- `UserPrincipalName`
- `whenChanged`
- `whenCreated`

After you add custom attributes in the LDAP connection settings, the **LDAP attribute** to receive drop-down list in the collector automatically includes the new attributes. Custom attributes are identified by a question mark next to the attribute name. If you added the same attribute for multiple domains, the attribute is listed only once in the drop-down list. You can view the domains by moving your cursor over the question mark. Domain names are displayed as links. If you click a link, the domain is automatically added to **LDAP accounts mapping** if it was not previously added.

If you deleted a custom attribute in the LDAP connection settings, manually delete the row containing the attribute from the mapping table in the collector. Account attribute information in KUMA is updated each time you import accounts.

2. Import accounts.

3. In the collector, in the **LDAP mapping** table, define the rules for mapping KUMA fields to LDAP attributes.

4. Restart the collector.

   After the collector is restarted, KUMA begins enriching events with accounts.

13. Select the **Disabled** check box if you do not want to use this LDAP connection.

    This check box is cleared by default.

14. Click the **Save** button.

    The LDAP connection to Active Directory will be created and displayed in the **LDAP server integration by tenant** window.

Account information from Active Directory will be requested immediately after the connection is saved, and then it will be updated at the specified frequency.

If you want to use multiple LDAP connections simultaneously for one tenant, you need to make sure that the domain controller address indicated in each of these connections is unique. Otherwise KUMA lets you enable only one of these connections. When checking the domain controller address, the program does not check whether the port is unique.

Creating a copy of an LDAP server connection

You can create an LDAP connection by copying an existing connection. In this case, all settings of the original connection are duplicated in the newly created connection.

To copy an LDAP connection:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the tenant for which you want to copy an LDAP connection.

   The **LDAP server integration** window opens.

4. Select the relevant connection.

5. In the opened **Connection parameters** window, click the **Duplicate connection** button.

   The New Connection window opens. The word `copy` will be added to the connection name.

6. If necessary, change the relevant settings.

7. Click the **Save** button.

   The new connection is created.

If you want to use multiple LDAP connections simultaneously for one tenant, you need to make sure that the domain controller address indicated in each of these connections is unique. Otherwise KUMA lets you enable only one of these connections. When checking the domain controller address, the program does not check whether the port is unique.

Changing an LDAP server connection

To change an LDAP server connection:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the tenant for which you want to change the LDAP server connection.

   The **LDAP server integration** window opens.

4. Click the LDAP server connection that you want to change.

   The window with the settings of the selected LDAP server connection opens.

5. Make the necessary changes to the settings.

6. Click the **Save** button.

   The LDAP server connection is changed. Restart the KUMA services that use LDAP server data enrichment for the changes to take effect.

Changing the data update frequency

KUMA queries the LDAP server to update account data. This occurs:

- Immediately after creating a new connection.
- Immediately after changing the settings of an existing connection.
- According to a regular schedule every several hours. Every 12 hours by default.
- Whenever a user creates a task to update account data.

When querying LDAP servers, a task is created in the **Task manager** section of the KUMA Console.

To change the schedule of KUMA queries to LDAP servers:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the relevant tenant.

   The **LDAP server integration** window opens.

4. In the **Data refresh interval** field, specify the required frequency in hours. The default value is 12.

The query schedule has been changed.

Changing the data storage period

Received user account data is stored in KUMA for 90 days by default if information about these accounts is no longer received from the Active Directory server. After this period, the data is deleted.

After KUMA account data is deleted, new and existing events are no longer enriched with this information. Account information will also be unavailable in alerts. If you want to view information about accounts throughout the entire period of alert storage, you must set the account data storage period to be longer than the alert storage period.

To change the storage period for the account data:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the relevant tenant.

   The **LDAP server integration** window opens.

4. In the **Data storage time** field, specify the number of days you need to store data received from the LDAP server.

The account data storage period is changed.

Starting account data update tasks

After a connection to an Active Directory server is created, tasks to obtain account data are created automatically. This occurs:

- Immediately after creating a new connection.
- Immediately after changing the settings of an existing connection.
- According to a regular schedule every several hours. Every 12 hours by default. The schedule can be changed.

Account data update tasks can be created manually. You can download data for all connections or for one connection of the required tenant.

To start an account data update task for all LDAP connections of a tenant:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the relevant tenant.

   The **LDAP server integration** window opens.

4. Click the **Import accounts** button.

A task to receive account data from the selected tenant is added to the **Task manager** section of the KUMA Console.

To start an account data update task for one LDAP connection of a tenant:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the relevant tenant.

   The **LDAP server integration** window opens.

4. Select the relevant LDAP server connection.

   The **Connection parameters** window opens.

5. Click the **Import accounts** button.

A task to receive account data from the selected connection of the tenant is added to the **Task manager** section of the KUMA Console.

Deleting an LDAP server connection

To delete LDAP connection to Active Directory:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Integrations**, click **LDAP server**.

   The **LDAP server integration by tenant** window opens.

3. Select the tenant that owns the relevant LDAP connection.

   The **LDAP server integration** window opens.

4. Click the LDAP connection that you want to delete and click the **Delete** button.

5. Confirm deletion of the connection.

The LDAP connection to Active Directory will be deleted.

Kaspersky Industrial CyberSecurity for Networks integration

Kaspersky Industrial CyberSecurity for Networks (hereinafter referred to as "KICS for Networks") is an application designed to protect the industrial enterprise infrastructure from information security threats, and to ensure uninterrupted operation. The application analyzes industrial network traffic to identify deviations in the values of process parameters, detect signs of network attacks, and monitor the operation and current state of network devices.

KICS for Networks version 4.0 or later can be integrated with KUMA. After configuring integration, you can perform the following tasks in KUMA:

- Import asset information from KICS for Networks to KUMA.
- Send asset status change commands from KUMA to KICS for Networks.

Unlike KUMA, KICS for Networks refers to assets as devices.

The integration of KICS for Networks and KUMA must be configured in both applications:

1. In KICS for Networks, you need to create a KUMA connector and save the communication data package of this connector.
2. In KUMA, the communication data package of the connector is used to create a connection to KICS for Networks.

The integration described in this section applies to importing asset information. KICS for Networks can also be configured to send events to KUMA. To do so, you need to create a SIEM/Syslog connector in KICS for Networks, and configure a collector on the KUMA side.

Configuring integration in KICS for Networks

The program supports integration with KICS for Networks version 4.0 or later.

It is recommended to configure integration of KICS for Networks and KUMA after ending Process Control rules learning mode. For more details, please refer to the documentation on KICS for Networks.

On the KICS for Networks side, integration configuration consists of creating a *KUMA-type connector*. In KICS for Networks, connectors are specialized application modules that enable KICS for Networks to exchange data with recipient systems, including KUMA. For more details on creating connectors, please refer to the documentation on KICS for Networks.

When a connector is added to KICS for Networks, a *communication data package* is automatically created for this connector. This is an encrypted configuration file for connecting to KICS for Networks that is used when configuring integration on the KUMA side.

Configuring integration in KUMA

It is recommended to configure integration of KICS for Networks and KUMA after ending Process Control rules learning mode. For more details, please refer to the documentation on KICS for Networks.

To configure integration with KICS for Networks in KUMA:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select a tenant for which you want to create an integration with KICS for Networks.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **KICS/KATA import** tab, click the **Add** button.
5. This opens the **Create connection** window.
6. To enable or disable the connection, turn on or turn off the **State** toggle switch
7. Click the plus sign icon (+) and select the communication data package that was created in KICS for Networks.
8. In the **Communication data package password** field, enter the password of the communication data package.
9. In the **API version** field, enter the KICS/KATA API version that will be used to communicate with KICS/KATA. By default, version 4 is used.
10. Click the **Add** button.
11. Turn on the **Enable response actions** toggle switch if you want to change the statuses of KICS for Networks assets by using KUMA response rules.
12. Click **Save**.

Integration with KICS for Networks is configured in KUMA, and the window shows the IP address of the node where the KICS for Networks connector will be running and its ID.

Enabling and disabling integration with KICS for Networks

To enable or disable KICS for Networks integration for a tenant:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant into which you want to import KICS/KATA assets.
   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **KICS/KATA import** tab, enable or disable the **State** toggle switch.
5. Click **Save**.

Changing the data update frequency

KUMA queries KICS for Networks to update its asset information. This occurs:

- Immediately after creating a new integration.
- Immediately after changing the settings of an existing integration.
- According to a regular schedule every several hours. This occurs every 3 hours by default.
- Whenever a user creates a task for updating asset data.

When querying KICS for Networks, a task is created in the **Task manager** section of the OSMP Console.

To edit the schedule for importing information about KICS for Networks assets:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant into which you want to import KICS/KATA assets.
   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **KICS/KATA import** tab, in the **Import interval (hours)** field, specify the time interval at which OSMP updates data about KICS/KATA assets.

The import schedule has been changed.

Special considerations when importing asset information from KICS for Networks

Importing assets

Assets are imported according to the asset import rules. Only assets with the **Authorized** and **Unauthorized** statuses are imported.

KICS for Networks assets are identified by a combination of the following parameters:

- IP address of the KICS for Networks instance with which the integration is configured.
- KICS for Networks connector ID is used to configure the integration.
- ID assigned to the asset (or "device") in the KICS for Networks instance.

Importing vulnerability information

When importing assets, KUMA also receives information about active vulnerabilities in KICS for Networks. If a vulnerability has been flagged as Remediated or Negligible in KICS for Networks, the information about this vulnerability is deleted from KUMA during the next import.

Information about asset vulnerabilities is displayed in the localization language of KICS for Networks in the **Asset details** window in the **Vulnerabilities** settings block.

> In KICS for Networks, vulnerabilities are referred to as risks and are divided into several types. All types of risks are imported into KUMA.

Imported data storage period

If information about a previously imported asset is no longer received from KICS for Networks, the asset is deleted after 30 days.

Changing the status of a KICS for Networks asset

After configuring integration, you can change the statuses of KICS for Networks assets from KUMA. Statuses can be changed either automatically or manually.

Asset statuses can be changed only if you enabled a response in the settings for connecting to KICS for Networks.

Manually changing the status of a KICS for Networks asset

Users with the Main administrator, Administrator, and Analyst roles in the tenants available to them can manually change the statuses of assets imported from KICS for Networks.

To manually change a KICS for Networks asset status:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Detection and response**.

2. Click the asset that you want to edit.

   The **Asset details** area opens in the right part of the window.

3. In the **Status in KICS for Networks** drop-down list, select the status that you need to assign to the KICS for Networks asset. The *Authorized* or *Unauthorized* statuses are available.

The asset status is changed. The new status is displayed in KICS for Networks and in KUMA.

Automatically changing the status of a KICS for Networks asset

Automatic changes to the statuses of KICS for Networks assets are implemented using response rules. The rules must be added to the correlator, which will determine the conditions for triggering these rules.

Integration with Neurodat SIEM IM

Neurodat SIEM IM is an information security monitoring system.

You can configure the export of KUMA events to Neurodat SIEM IM. Based on incoming events and correlation rules, Neurodat SIEM IM automatically generates information security incidents.

To configure integration with Neurodat SIEM IM:

1. Connect to the Neurodat SIEM IM server over SSH using an account with administrative privileges.

2. Create a backup copy of the /opt/apache-tomcat-<server version>/conf/neurodat/soz_settings.properties configuration file.

3. In the /opt/apache-tomcat-<server version>/conf/neurodat/soz_settings.properties configuration file, edit the following settings as follows:

   - `kuma.on=true`

     This setting is an attribute of Neurodat SIEM IM interaction with KUMA.

   - `job_kuma=com.cbi.soz.server.utils.scheduler.KumaIncidentsJob`

   - `jobDelay_kuma=5000`

   - `jobPeriod_kuma=60000`

4. Save changes of the configuration file.

5. Run the following command to restart the tomcat service:

   ```
   sudo systemctl restart tomcat
   ```

6. Obtain a token for the user in KUMA. To do so:

   a. In the main menu, go to **Settings** → **API tokens**.

      This opens the **API tokens** window.

   b. Click the **Add token** button.

      The **Add token** window opens.

   c. If necessary, set the token expiration date:

      - Select the **No expiration date** check box.
      - In the **Expiration date** field, use the calendar to specify the date and time when the created token will expire.

   d. Click the **Generate** button.

      The **Token** field with an automatically generated token is displayed in the user details area. Copy it.

      When the window is closed, the token is no longer displayed. If you did not copy the token before closing the window, you will have to generate a new token.

   e. Click **Save**.

7. Log in to Neurodat SIEM IM using the 'admin' account or another account that has the Administrator role for the organization you are configuring or the Administrator role for all organizations.

8. In the **Administration** → **Organization structure** menu item, select or create an organization that you want to receive incidents from KUMA.

9. On the organization form, do the following:

   a. Select the **Configure integration with KUMA** check box.

   b. In the **KUMA IP address and port** field, specify the KUMA API address, for example, `https://192.168.58.27:7223/api/v1/`.

   c. In the **KUMA API key** field, specify the user token obtained at step 6.

   d. Save the organization information.

Integration with KUMA is configured.

Neurodat SIEM IM tests access to KUMA and, if successful, displays a message about being ready to receive data from KUMA.

Kaspersky Automated Security Awareness Platform

Kaspersky Automated Security Awareness Platform (hereinafter also referred to as "KASAP") is an online learning platform that allows users to learn the rules of information security and threats related to it in their daily work, as well as to practice using real examples.

KASAP can be integrated with KUMA. After configuring integration, you can perform the following tasks in KUMA:

- Change user learning groups.
- View information about the courses taken by the users and the certificates they received.

Integration between KASAP and KUMA includes configuring API connection to KASAP. The process takes place in both solutions:

1. In KASAP, create an authorization token and obtain an address for API requests.
2. In KUMA, specify the address for API requests in KASAP, add an authorization token for API requests, and specify the email address of the KASAP administrator to receive notifications.

Creating a token in KASAP and getting a link for API requests

In order to be authorized, the API requests from KUMA to KASAP must be signed by a token created in KASAP. Only the company administrators can create tokens.

Creating a token

To create a token:

1. Sign in to the KASAP web interface.
2. In the **Control panel** section, click the **Import and synchronization** button, and then open the **Open API** tab.
3. Click the **New token** button and select the API methods used for integration in the window that opens:
   - GET /openapi/v1/groups
   - POST /openapi/v1/report
   - PATCH /openapi/v1/user/:userid
4. Click the **Generate token** button.
5. Copy the token and save it in any convenient way. This token is required to configure integration in KUMA.

> The token is not stored in the KASAP system in the open form. After you close the **Get token** window, the token is unavailable for viewing. If you close the window without copying the token, you will need to click the **New token** button again for the system to generate a new token.

The issued token is valid for 12 months. After this period, the token is revoked. The issued token is also revoked if it is not used for 6 months.

Getting a link for API requests

To get the link used in KASAP for API requests:

1. Log in to the KASAP platform console.
2. In the **Control panel** section, click the **Import and synchronization** button, and then open the **Open API** tab.
3. A link for accessing KASAP using the Open API is located at the bottom part of the window. Copy the link and save it in any convenient way. This link is required to configure integration in KUMA.

Configuring integration in KUMA

To configure KUMA integration with KASAP:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the **KUMA settings** window.
2. Under **Integrations**, click **Kaspersky Automated Security Awareness Platform**.

   The **Kaspersky Automated Security Awareness Platform** window opens.
3. In the **Secret** field click the plus icon (+) to create a secret of the **token** by entering the token received from KASAP:

   a. In the **Name** field, enter the name of the secret. Must contain 1 to 128 Unicode characters.

   b. In the **Token** field, enter the authorization token for API requests to KASAP.

   c. If necessary, add the secret description in the **Description** field.

   d. Click **Save**.
4. In the **KASAP Open API URL** field, specify the address used by KASAP for API requests.
5. In the **KASAP administrator email** field, specify the email address of the KASAP administrator who receives notifications when users are added to the learning groups using KUMA.
6. If necessary, in the **Proxy** drop-down list select the proxy server resource to be used to connect to KASAP.
7. To disable or enable integration with KASAP, select or clear the **Disabled** check box.
8. Click **Save**.

Integration with KASAP is configured in KUMA. When viewing information about alerts and incidents, you can select associated users to view which learning courses they have taken and to change their learning group.

Viewing information about the users from KASAP and changing learning groups

After configuring the integration between KASAP and KUMA, the following information from KASAP becomes available in alerts and incidents when you view data about associated users:

- The learning group to which the user belongs.
- The trainings passed by the user.
- The planned trainings and the current progress.
- The received certificates.

To view data about the user from KASAP:

1. In OSMP Console, in the **Alerts** or **Incidents** section, select the relevant alert or incident.
2. In the **Related users** section, click the desired account.
   The **Account details** window opens on the right side of the screen.
3. Select the **KASAP courses details** tab.

   The window displays information about the user from KASAP.

You can change the learning group of a user in KASAP.

To change a user learning group in KASAP:

1. In OSMP Console, in the **Alerts** or **Incidents** section, select the relevant alert or incident.
2. In the **Related users** section, click the desired account.
   The **Account details** window opens on the right side of the screen.
3. In the **Assign KASAP group** drop-down list, select the KASAP learning group you want to assign the user to.
4. Click **Apply**.

   The user is moved to the selected KASAP group, the KASAP company administrator is notified of the change in the learning group, and the study plan is recalculated for the selected learning group.

For details on learning groups and how to get started, refer to the KASAP documentation.

Sending notifications to Telegram

This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
The terms and conditions of premium technical support do not apply to this integration; support requests are processed without a guaranteed response time.

You can configure sending notifications to Telegram when KUMA correlation rules are triggered. This can reduce the response time to threats and, if necessary, make more persons informed.

Configure Telegram notifications involves the following steps:

1. Creating and configuring a Telegram bot

   A special bot sends notifications about triggered correlation rules. It can send notifications to a private or group Telegram chat.

2. Creating a script for sending notifications

   You must create a script and save it on the server where the correlator is installed.

3. Configuring notifications in KUMA

   Configure a KUMA response rule that starts a script to send notifications and add this rule to the correlator.

Creating and configuring a Telegram bot

To create and configure a Telegram bot:

1. In the Telegram application, find the BotFather bot and open a chat with it.

2. In the chat, click **Start**.

3. Create a new bot using the following command:

   ```
   /newbot
   ```

4. Enter the name of the bot.

5. Enter the login name of the bot.

   The bot is created. You receive a link to the chat that looks like t.me/<bot login> and a token for contacting the bot.

6. If you want to use the bot in a group chat, and not in private messages, edit privacy settings:

   a. In the BotFather chat, enter the command:

   ```
   /mybots
   ```

   b. Select the relevant bot from the list.

c. Click **Bot Settings** → **Group Privacy** and select **Turn off**.

The bot can now send messages to group chats.

7. To open a chat with the bot you created, use the t.me/<botlogin> link that you obtained at step 5, and click **Start**.

8. If you want the bot to send private messages to the user:

   a. In the chat with the bot, send any message.

   b. Follow the https://t.me/getmyid_bot link and click **Start**.

   c. The response contains the `Current chat ID`. You need this value to configure the sending of messages.

9. If you want the bot to send messages to the group chat:

   a. Add https://t.me/getmyid_bot to the group chat for receiving notifications from KUMA.

   The bot sends a message to the group chat, the message contains the `Current chat ID` value. You need this value to configure the sending of messages.

   b. Remove the bot from the group.

10. Send a test message through the bot. To do so, paste the following link into the address bar of your browser:

    `https://api.telegram.org/bot<token>/sendMessage?chat_id=<chat_id>&text=test`

    where `<token>` is the value obtained at step 5, and `<chat_id>` is the value obtained at step 8 or 9.

As a result, a test message should appear in the personal or group chat, and the JSON in the browser response should be free of errors.

Creating a script for sending notifications

To create a script:

1. In the console of the server on which the correlator is installed, create a script file and add the following lines to it:

```
#!/bin/bash
set -eu
CHAT_ID=<Current chat ID value obtained at step 8 or 9 of the Telegram bot setup instructions>
TG_TOKEN=<token value obtained at step 5 of the Telegram bot setup instructions>
RULE=$1
TEXT="<b>$RULE</b> rule triggered."
curl --data-urlencode "chat_id=$CHAT_ID" --data-urlencode "text=$TEXT" --data-urlencode "parse_mode=HTML" https://api.telegram.org/bot$TG_TOKEN/sendMessage
```

If the correlator server does not have Internet access, you can use a proxy server:

```
#!/bin/bash
set -eu
CHAT_ID=<Current chat ID value obtained at step 8 or 9 of the Telegram bot setup instructions>
TG_TOKEN=<token value obtained at step 5 of the Telegram bot setup instructions>
RULE=$1
TEXT="<b>$RULE</b> rule triggered."
PROXY=<address and port of the proxy server>
```

```
curl --proxy $PROXY --data-urlencode "chat_id=$CHAT_ID" --data-urlencode "text=$TEXT"
--data-urlencode "parse_mode=HTML" https://api.telegram.org/bot$TG_TOKEN/sendMessage
```

2. Save the script to the correlator directory at /opt/kaspersky/kuma/correlator/<`ID of the correlator that must respond to events`>/scripts/.

   For information about obtaining the correlator ID, see the Getting service identifier section.

3. Make the 'kuma' user the owner of the file and grant execution rights:

```
chown kuma:kuma /opt/kaspersky/kuma/correlator/<ID of the correlator that must
respond>/scripts/<script name>.sh
chmod +x /opt/kaspersky/kuma/correlator/<ID of the correlator that must
respond>/scripts/<script name>.sh
```

Configuring notifications in KUMA

To configure the sending of KUMA notifications to Telegram:

1. Create a response rule:

   a. In the KUMA Console, select the **Resources** → **Response rules** section and click **Add response rule**.

   b. This opens the **Create response rule** window; in that window, in the **Name** field, enter the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

   d. In the **Type** drop-down list, select **Run script**.

   e. In the **Script name** field, enter the name of the script.

   f. In the **Script arguments** field, enter `{{.Name}}`.

   This passes the name of the correlation event as the argument of the script.

   g. Click **Save**.

2. Add the response rule to the correlator:

   a. In the main menu, go to **Monitoring** → **Resources and services** → **Correlators**, and then select the correlator in whose folder you placed the created script for sending notifications.

   b. In the steps tree, select **Response rules**.

   c. Click **Add**.

   d. In the **Response rule** drop-down list, select the rule added at step 1 of these instructions.

   e. In the steps tree, select **Setup validation**.

   f. Click the **Save and restart services** button.

   g. Click the **Save** button.

Sending notifications about triggered KUMA rules to Telegram is configured.

UserGate integration

This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
The terms and conditions of premium technical support do not apply to this integration; support requests are processed without a guaranteed response time.

UserGate is a network infrastructure security solution that protects personal information from the risks of external intrusions, unauthorized access, viruses, and malware.

Integration with UserGate allows automatically blocking threats by IP address, URL, or domain name whenever KUMA response rules are triggered.

Configuring the integration involves the following steps:

1. Configuring integration in UserGate
2. Preparing a script for the response rule
3. Configuring the KUMA response rule

Configuring integration in UserGate

To configure integration in UserGate:

1. Connect to the UserGate web interface under an administrator account.
2. Go to **UserGate** → **Administrators** → **Administrator profiles**, and click **Add**.
3. In the **Profile settings** window, specify the profile name, for example, `API`.
4. On the **API Permissions** tab, add read and write permissions for the following objects:
   - content
   - core
   - firewall
   - nlists
5. Click **Save**.
6. In the **UserGate** → **Administrators** section, click **Add** → **Add local administrator**.
7. In the **Administrator properties** window, specify the login and password of the administrator.
   In the **Administrator profile** field, select the profile created at step 3.
8. Click **Save**.

9. In the address bar of your browser, after the address and port of UserGate, add `?features=zone-xml-rpc` and press **ENTER**.

10. Go to the **Network** → **Zones** section and for the zone of the interface that you want to use for API interaction, go to the **Access Control** tab and select the check box next to the **XML-RPC for management** service.

    If necessary, you can add the IP address of the KUMA correlator whose correlation rules must trigger blocking in UserGate, to the list of allowed addresses.

11. Click **Save**.

Preparing a script for integration with UserGate

To prepare a script for use:

1. Copy the ID of the correlator whose correlation rules you want to trigger blocking of URL, IP address, or domain name in UserGate:

   a. In the KUMA Console, go to the **Resources** → **Active services**.

   b. Select the check box next to the correlator whose ID you want to obtain, and click **Copy ID**.

      The correlator ID is copied to the clipbоard.

2. Download the script:

   https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

3. Open the script file and in the **Enter UserGate Parameters** section, in the **login** and **password** parameters, specify the credentials of the UserGate administrator account that was created at step 7 of configuring the integration in UserGate.

4. Place the downloaded script on the KUMA correlator server at the following path: /opt/kaspersky/kuma/correlator/<`correlator ID from step 1`>/scripts/.

5. Connect to the correlator server via SSH and go to the path from step 4:

   ```
   cd /opt/kaspersky/kuma/correlator/< correlator ID from step 1 >/scripts/
   ```

6. Run the following command:

   ```
   chmod +x ug.py && chown kuma:kuma ug.py
   ```

The script is ready to use.

Configuring a response rule for integration with UserGate

To configure a response rule:

1. Create a response rule:

   a. In the KUMA Console, select the **Resources** → **Response rules** section and click **Add response rule**.

   b. This opens the **Create response rule** window; in that window, in the **Name** field, enter the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

d. In the **Type** drop-down list, select **Run script**.

e. In the **Script name** field, enter the name of the script. `ug.py`.

f. In the **Script arguments** field, specify:

- one of the operations depending on the type of the object being blocked:
  - `blockurl` to block access by URL
  - `blockip` to block access by IP address
  - `blockdomain` to block access by domain name
- `-i {{< KUMA field from which the value of the blocked object must be taken, depending on the operation >}}`

  Example:
  `blockurl -i {{.RequetstUrl}}`

g. In the **Conditions** section, add conditions corresponding to correlation rules that require blocking in UserGate when triggered.

h. Click **Save**.

2. Add the response rule to the correlator:

   a. In the **Resources** → **Correlators** section, select the correlator that must respond and in whose directory you placed the script.

   b. In the steps tree, select **Response rules**.

   c. Click **Add**.

   d. In the **Response rule** drop-down list, select the rule added at step 1 of these instructions.

   e. In the steps tree, select **Setup validation**.

   f. Click **Save and reload services**.

   g. Click the **Save** button.

The response rule is linked to the correlator and ready to use.

Integration with Kaspersky Web Traffic Security

This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
The terms and conditions of premium technical support do not apply to this integration; support requests are processed without a guaranteed response time.

You can configure integration with the Kaspersky Web Traffic Security web traffic analysis and filtering system (hereinafter also referred to as "KWTS").

Configuring the integration involves creating KUMA response rules that allow running KWTS tasks. Tasks must be created in advance in the KWTS web interface.

Configuring the integration involves the following steps:

1. Configuring integration in KWTS
2. Preparing a script for the response rule
3. Configuring the KUMA response rule

Configuring integration in KWTS

To prepare the integration in KWTS:

1. Connect to the KWTS web interface under an administrator account and create a role with permissions to view and create/edit a rule.

   For more details on creating a role, see the *Kaspersky Web Traffic Security Help*.

2. Assign the created role to a user with NTML authentication.

   You can use a local administrator account instead.

3. In the **Rules** section, go to the **Access** tab and click **Add rule**.
4. In the **Action** drop-down list, select **Block**.
5. In the **Traffic filtering** drop-down list, select the **URL** value, and in the field on the right, enter a nonexistent or known malicious address.
6. In the **Name** field, enter the name of the rule.
7. Enable the rule using the **Status** toggle switch.
8. Click **Add**.
9. In the KWTS web interface, open the rule you just created.
10. Make a note of the ID value that is displayed at the end of the page address in the browser address bar.

    You must use this value when configuring the response rule in KUMA.

The integration is prepared on the KWTS side.

Preparing a script for integration with KWTS

To prepare a script for use:

1. Copy the ID of the correlator whose correlation rules you want to trigger blocking of URL, IP address, or domain name in KWTS:

   a. In the KUMA Console, go to the **Resources** → **Active services**.

   b. Select the check box next to the correlator whose ID you want to obtain, and click **Copy ID**.

   The correlator ID is copied to the clipboard.

2. Download the script and library:

   https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

3. Place the downloaded script on the KUMA correlator server at the following path: /opt/kaspersky/kuma/correlator/<`correlator ID from step 1`>/scripts/.

4. Connect to the correlator server via SSH and go to the path from step 3:

   ```
   cd /opt/kaspersky/kuma/correlator/< correlator ID from step 1 >/scripts/
   ```

5. Run the following command:

   ```
   chmod +x kwts.py kwtsWebApiV6.py && chown kuma:kuma kwts.py kwtsWebApiV6.py
   ```

The script is ready to use.

Configuring a response rule for integration with KWTS

To configure a response rule:

1. Create a response rule:

   a. In the KUMA Console, select the **Resources** → **Response rules** section and click **Add response rule**.

   b. This opens the **Create response rule** window; in that window, in the **Name** field, enter the name of the rule.

   c. In the **Tenant** drop-down list, select the tenant that owns the resource.

   d. In the **Type** drop-down list, select **Run script**.

   e. In the **Script name** field, enter the name of the script. kwts.py.

   f. In the **Script arguments** field, specify:

   - `--host` — address of the KWTS server.
   - `--username` — name of the user account created in KWTS or local administrator.
   - `--password` — KWTS user account password.

- `--rule_id` — ID of the rule created in KWTS.
- Specify one of the options depending on the type of the object being blocked:
  - `--url` — specify the field of the KUMA event from which you want to obtain the URL, for example, `{{.RequestUrl}}`.
  - `--ip` — specify the field of the KUMA event from which you want to obtain the IP address, for example, `{{.DestinationAddress}}`.
  - `--domain` — specify the field of the KUMA event from which you want to obtain the domain name, for example, `{{.DestinationHostName}}`.
- `--ntlm` — specify this option if the KWTS user was created with NTLM authentication.

  Example:
  ```
  --host <address> --username <user> --password <pass> --rule_id <id> --url {{.RequestUrl}}
  ```

g. In the **Conditions** section, add conditions corresponding to correlation rules that require blocking in KWTS when triggered.

h. Click **Save**.

2. Add the response rule to the correlator:

   a. In the **Resources** → **Correlators** section, select the correlator that must respond and in whose directory you placed the script.

   b. In the steps tree, select **Response rules**.

   c. Click **Add**.

   d. In the **Response rule** drop-down list, select the rule added at step 1 of these instructions.

   e. In the steps tree, select **Setup validation**.

   f. Click **Save and reload services**.

   g. Click the **Save** button.

The response rule is linked to the correlator and ready to use.

Integration with Kaspersky Secure Mail Gateway

This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
The terms and conditions of premium technical support do not apply to this integration; support requests are processed without a guaranteed response time.

You can configure integration with the Kaspersky Secure Mail Gateway mail traffic analysis and filtering system (hereinafter also referred to as "KSMG").

Configuring the integration involves creating KUMA response rules that allow running KSMG tasks. Tasks must be created in advance in the KSMG web interface.

Configuring the integration involves the following steps:

1. Configuring integration in KSMG
2. Preparing a script for the response rule
3. **Configuring the KUMA response rule**

Configuring integration in KSMG

To prepare the integration in KSMG:

1. Connect to the KSMG web interface under an administrator account and create a role with permissions to view and create/edit a rule.

   For more details on creating a role, see the *Kaspersky Secure Mail Gateway Help*.

2. Assign the created role to a user with NTML authentication.

   You can use the 'Administrator' local administrator account.

3. In the **Rules** section, click **Create**.
4. In the left pane, select the **General** section.
5. Enable the rule using the **Status** toggle switch.
6. In the **Rule name** field, enter the name of the new rule.
7. Under **Mode**, select one of the message processing options that meets the criteria of this rule.
8. Under **Sender** on the **Email addresses** tab, enter a nonexistent or known malicious sender address.
9. Under **Recipient** on the **Email addresses** tab, specify the relevant recipients or the "*" character to select all recipients.
10. Click the **Save** button.
11. In the KSMG web interface, open the rule you just created.
12. Make a note of the ID value that is displayed at the end of the page address in the browser address bar.

    You must use this value when configuring the response rule in KUMA.

The integration is prepared on the KSMG side.

Preparing a script for integration with KSMG

To prepare a script for use:

1. Copy the ID of the correlator whose correlation rules must trigger the blocking of the IP address or email address of the message sender in KSMG:

   a. In the KUMA Console, go to the **Resources** → **Active services**.

   b. Select the check box next to the correlator whose ID you want to obtain, and click **Copy ID**.

   The correlator ID is copied to the clipboard.

2. Download the script and library:

   https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

3. Place the downloaded script on the KUMA correlator server at the following path: /opt/kaspersky/kuma/correlator/<`correlator ID from step 1`>/scripts/.

4. Connect to the correlator server via SSH and go to the path from step 3:

   ```
   cd /opt/kaspersky/kuma/correlator/< correlator ID from step 1 >/scripts/
   ```

5. Run the following command:

   ```
   chmod +x ksmg.py ksmgWebApiV2.py && chown kuma:kuma ksmg.py ksmgWebApiV2.py
   ```

The script is ready to use.

Importing asset information from RedCheck

> This integration is an example and may require additional configuration depending on the versions used and the specifics of the infrastructure.
> The terms and conditions of premium technical support do not apply to this integration; support requests are processed without a guaranteed response time.

RedCheck is a system for monitoring and managing the information security of an organization.

You can import asset information from RedCheck network device scan reports into KUMA.

Import is available from simple "Vulnerabilities" and "Inventory" reports in CSV format, grouped by hosts.

Imported assets are displayed in OSMP Console in the **Asset management** → **Assets (Devices)** → **Detection and response** section. If necessary, you can edit the settings of assets.

Data is imported through the API using the redcheck-tool.py utility. The utility requires Python 3.6 or later and the following libraries:

- csv
- re

- json
- requests
- argparse
- sys

To import asset information from a RedCheck report:

1. Generate a network asset scan report in RedCheck in CSV format and copy the report file to the server where the script is located.

   For more details about scan tasks and output file formats, refer to the RedCheck documentation.

2. Create a file with the token for accessing the KUMA REST API.

   The account for which the token is created must satisfy the following requirements:

   - Administrator or Analyst role.
   - Access to the tenant into which the assets will be imported.
   - Rights to use API requests: GET /assets, GET /tenants, POST/assets/import.

3. Download the script:

   https://box.kaspersky.com/d/2dfd1d677c7547a7ac1e/

4. Copy the redcheck-tool.py tool to the server hosting the KUMA Core and make the tool's file executable:

   ```
   chmod +x <path to the redcheck-tool.py file>
   ```

5. Run the redcheck-tool.py utility:

   ```
   python3 redcheck-tool.py --kuma-rest <address and port of the KUMA REST API server> --token <API token> --tenant <name of the tenant in which the assets must be placed> --vuln-report <full path to the "Vulnerabilities" report file> --inventory-report <full path to the "Inventory" report file>
   ```

   Example:
   ```
   python3 --kuma-rest example.kuma.com:7223 --token 949fc03d97bad5d04b6e231c68be54fb --tenant Main --vuln-report /home/user/vuln.csv --inventory-report /home/user/inventory.csv
   ```

   You can use additional flags and commands for import operations. For example, the `-v` command displays an extended report on the received assets. A detailed description of the available flags and commands is provided in the "Flags and commands of redcheck-tool.py" table. You can also use the `--help` command to view information on the available flags and commands.

The asset information is imported from the RedCheck report to KUMA. The console displays information on the number of new and updated assets.

Example:

```
inventory has been imported for 2 host(s)
software has been imported for 5 host(s)
vulnerabilities has been imported for 4 host(s)
```

Example of extended import information:

```
[inventory import] Host: localhost Code: 200 Response: {'insertedIDs': {'0': '52ca11c6-a0e6-4dfd-8ef9-
bf58189340f8'}, 'updatedCount': 0, 'errors': []}
[inventory import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {'0': '1583e552-5137-4164-92e0-
01e60fb6edb0'}, 'updatedCount': 0, 'errors': []}
[software import][error] Host: localhost Skipped asset with FQDN localhost or IP 127.0.0.1
[software import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {}, 'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.2 Code: 200 Response: {'insertedIDs': {}, 'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.1 Code: 200 Response: {'insertedIDs': {'0': '0628f683-c20c-4107-abf3-
d837b3dbbf01'}, 'updatedCount': 0, 'errors': []}
[vulnerabilities import] Host: localhost Code: 200 Response: {'insertedIDs': {}, 'updatedCount': 1, 'errors': []}
[vulnerabilities import] Host: 10.0.0.3 Code: 200 Response: {'insertedIDs': {'0': 'ed01e0a8-dcb0-4609-ab2b-
91e50092555d'}, 'updatedCount': 0, 'errors': []}
inventory has been imported for 2 host(s)
software has been imported for 1 host(s)
vulnerabilities has been imported for 4 host(s)
```

The tool works as follows when importing assets:

- KUMA overwrites the data of assets imported through the API, and deletes information about their resolved vulnerabilities.
- KUMA skips assets with invalid data.

Flags and commands of redcheck-tool.py

Flags and commands	Mandatory	Description
`--kuma-rest < address and port of the KUMA server >`	Yes	Port 7223 is used for API requests by default. You can change the port if necessary.
`--token < token >`	Yes	The value of the option must contain only the token. The Administrator or Analyst role must be assigned to the user account for which the API token is being generated.
`--tenant < tenant name >`	Yes	Name of the KUMA tenant in which the assets from the RedCheck report will be imported.
`--vuln-report < full path to the "Vulnerabilities" report >`	Yes	"Vulnerabilities" report file in CSV format.
`--inventory-report < full path to the "Inventory" report file >`	No	"Inventory" report file in CSV format.
`-v`	No	Display extended information about the import of assets.

Possible errors

Error message	Description
Tenant %w not found	The tenant name was not found.
Tenant search error: Unexpected status Code: %d	An unexpected HTTP response code was received while searching for the tenant.
Asset search error: Unexpected status Code: %d	An unexpected HTTP response code was received while searching for an asset.
[%w import][error] Host: %w Skipped asset with FQDNlocalhost or IP 127.0.0.1	When importing inventory/vulnerabilities information, host cfqdn=localhost or ip=127.0.0.1 was skipped.

Configuring receipt of Sendmail events

You can configure the receipt of Sendmail mail agent events in the KUMA SIEM system.

Configuring event receiving consists of the following steps:

1. Configuring Sendmail logging.
2. Configuring the event source server.
3. Creating a KUMA collector.

   To receive Sendmail events, use the following values in the Collector Installation Wizard:

   - At the **Event parsing** step, select the **[OOTB] Sendmail syslog** normalizer.
   - At the **Transport** step, select the **tcp** or **udp** connector type.
4. Installing KUMA collector.
5. Verifying receipt of Sendmail events in the KUMA collector.

   You can verify that the Sendmail event source server is correctly configured in the Searching for related events section of OSMP Console.

Configuring Sendmail logging

By default, events of the Sendmail system are logged to syslog.

To make sure that logging is configured correctly:

1. Connect via SSH to the server on which the Sendmail system is installed.
2. Run the following command:

   ```
   cat /etc/rsyslog.d/50-default.conf
   ```

   The command should return the following string:

   ```
   mail.* -/var/log/mail.log
   ```

If logging is configured correctly, you can proceed to configuring the export of Sendmail events.

Configuring export of Sendmail events

Events are sent from the Sendmail mail agent server to the KUMA collector using the rsyslog service.

To configure transmission of Sendmail events to the collector:

1. Connect to the server where Sendmail is installed using an account with administrative privileges.
2. In the /etc/rsyslog.d/ directory, create the Sendmail-to-siem.conf file and add the following line to it:

   ```
   If $programname contains 'sendmail' then @< <IP address of the collector> : <port of the collector> >
   ```

   Example:

   ```
   If $programname contains 'sendmail' then @192.168.1.5:1514
   ```

   If you want to send events via TCP, the contents of the file must be as follows:

```
If $programname contains 'sendmail' then @@< <IP address of the collector> : <port of
the collector> >
```

3. Create a backup copy of the /etc/rsyslog.conf file.

4. Add the following lines to the /etc/rsyslog.conf configuration file:

   ```
   $IncludeConfig /etc/Sendmail-to-siem.conf
   $RepeatedMsgReduction off
   ```

5. Save your changes.

6. Restart the rsyslog service by executing the following command:

   ```
   sudo systemctl restart rsyslog.service
   ```

Managing KUMA

This section describes KUMA general settings.

Configuring the KUMA Core

To manage KUMA Core settings:

1. In the main application window, go to the **Settings** → **KUMA** → **Open KUMA settings** section.

   This opens the KUMA settings window.

2. Under **Other**, click **Common**.

   This opens the **Common** window.

3. In the **Core settings** section, specify the following settings:

   - Use the **Debug** toggle switch to enable or disable system operation logging (debug mode). The toggle switch is turned off by default.

   - If debug mode is enabled, you can use the **Dump KSC HTTP Traffic** toggle switch to enable or disable the ability to dump HTTP traffic of Kaspersky Security Center.

   - If the dumping of HTTP traffic of Kaspersky Security Center is enabled, you can use the **Periodic dumping** toggle switch to enable or disable the dumping of Kaspersky Security Center HTTP requests and responses to the event log.

     After enabling the **Enable periodic dump** check box, specify the dump settings under **Dump settings**.

   - In the **Week start day** drop-down list, select the first day of the week to be used for the entire KUMA instance. The selected value is applied when selecting periods and specifying time intervals in all sections of KUMA. The default is **Monday**. Only a user with the General Administrator role can modify this setting. The selected value is saved in the KUMA Core settings.

   - If necessary, click the **Reissue** button to reissue the AI service certificate and the internal CA certificate.

     > After reissuing the AI service certificate, you must manually restart all services. To apply the new AI service certificate, you must restart the KUMA Core.

4. If you do not want to use a self-signed KUMA certificate, in the **External TLS pair** section, upload a certificate and key for the TLS pair.

   By clicking the **Update certificate** button, you can update the certificate for the TLS pair.

5. Under **Core properties**, specify the following settings:

   - Select the **Compacting SQLite** check box and specify the amount of disk space in GB. This allows you to reduce the size of the database for more efficient use of disk space.

     By default, the check box is selected, and the disk space is set to 10,000 GB.

   - In the **Activation tag** field, specify an applicable tag.

   - In the **Alert retention period, days** field, specify the retention period for alerts in days. You can specify any value greater than 0.

     The default retention period is 365 days.

- In the **Asset bulk merge size** field, specify the number of assets in one group (batch) into which the total set of assets is split before the start of the merge. Each group of assets is sequentially processed by the program: assets are merged, verified and saved. If a value of 10 is specified, then 100 assets are split into 10 groups of 10 assets, and the groups are processed one by one. Specifying a smaller value yields smaller groups, which makes the merge faster because the application processes less data per step.

  You can specify any value greater than 0. The default setting is 10.

- In the **Maximum number of services to be started simultaneously** field, specify the number of services that KUMA can start simultaneously.

  The default setting is 10.

  > Too many services running at the same time can slow down the processing of KUMA data.

- In the **Correlation rule exclusions TTL, hours** field, specify the lifetime in hours for correlation rule exclusions. You can specify a value from 1 to 8760 (24*365) inclusive.

  The default setting is 72.

- In the **Disk buffer size of service destination of Core (for sending events), MB** field, specify the size of the disk buffer on the Core that is allocated for each service. This buffer is used when internal events cannot be sent to correlators. You can set any buffer size greater than 0.

  The default setting is 1024 MB.

- In the **Timeout of sending events via service destination of Core, seconds** field, specify the timeout for establishing the connection to the service when sending events. You can specify any timeout longer than 0 seconds.

  The default setting is 15 seconds.

- In the **The hour of the day when Core is authorized to perform self-service tasks** field, specify the time (in UTC) when the Core can perform self-service operations. The Core performs such operations on a daily basis. You can specify a value from 0 to 23, inclusive.

  > We recommend performing self-service operations during nighttime, when users are not using the functionality of the Core.

  The default setting is 1 (UTC time).

- In the **Max stored file size, MB** field, specify the maximum size of the compressed data of the downloaded GeoIP database and repository packages.

  The default setting is 300 MB.

- In the **Resource history retention period, days** field, specify the retention period for the resource changes history in days. You can set any value greater than or equal to 0.

  The default setting is 12 days.

- In the **Cluster request timeout from the UI, seconds** field, specify how long the KUMA web client must wait for a response after requesting the list of storages. You can set any value greater than 0.

  The default setting is 30 seconds.

- In the **Storage nodes health check timeout, seconds** field, specify the time to wait for a response after requesting an availability check of certain storage nodes. You can set any value greater than 0.

  The default setting is 30 seconds.

- If necessary, in the **KUMA UI balancer hostname and port** field, specify the host name and port of the KUMA user interface balancer.

  Given the distributed nature of the KUMA Core, multiple Core nodes can be configured to work as a cluster. You can configure the connection to the cluster through the balancer.

  > You can specify only one balancer URL.

  The balancer URL must be specified in the `hostname:port` format, where

  - `hostname` is the hostname of the balancer.
  - `port` is the port of the balancer.

  > Other balancer URL formats are not allowed.

6. Click **Save**.

Proxies

Proxy servers are used to store proxy server configuration settings, for example, in destinations. The http type is supported.

Available settings:

- **Name** (required)—unique name of the proxy server. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Secret separately**—if this check box is selected, the window displays the required **URL** field in which you can specify the connection URL, and a **Secret** drop-down list with secrets of the 'credentials' type. In this way, you can view connection information without having to re-create a large number of connections if the password of the user account that you used for the connections changes. If the check box is cleared, the **URL** and **Secret** fields are not available. This check box is cleared by default.
- **URL** (required) is the field for specifying the URL of the connection. It is used together with a secret of the 'credentials' type. Available if the **Secret separately** check box is selected.
- **Secret** is a drop-down list for selecting an existing secret or creating a new secret of the 'credentials' type. The drop-down list is available if the **Secret separately** check box is selected.
- **Use URL from the secret** (required)—drop-down list to select a secret resource that stores URLs of proxy servers. If required, a secret can be created in the proxy server creation window by clicking the plus icon (+). The selected secret can be changed by clicking the pencil icon (✎).

- **Do not use for domains**—one or more domains that require direct access.
- **Description**—up to 4,000 Unicode characters.

Connecting to an SMTP server

KUMA can be configured to send email notifications using an SMTP server. Users will receive notifications if the **Receive email notifications** check box is selected in their profile settings.

Only one SMTP server can be added to process KUMA notifications. An SMTP server connection is managed in OSMP Console under **Settings** → **KUMA** → **Open KUMA settings** → **Common** → **SMTP server settings**.

To configure SMTP server connection:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.
2. This opens the KUMA settings window.
3. Under **Other**, click **Common**.
4. This opens the **Common** window.
5. In the **SMTP server settings** block, change the relevant settings:
   - **Disabled**—select this check box if you want to disable connection to the SMTP server.
   - **Host** (required)—SMTP host in one of the following formats: hostname, IPv4, IPv6.
   - **Port** (required)—SMTP port. The value must be an integer from 1 to 65,535.
   - **From** (required)—email address of the message sender. For example, `kuma@company.com`.
   - **Alias for KUMA Core server**—name of the KUMA Core server that is used in your network. Must be different from the FQDN.
   - If necessary, use the **Secret** drop-down list to select a secret of the **credentials** type that contains the account credentials for connecting to the SMTP server.

     Add secret

     To create a secret:

     1. In the **Name** field, enter the name of the secret.
     2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector.
     3. If necessary, enter a description of the secret in the **Description** field.
     4. Click the **Create** button.

     The secret is added and displayed in the **Secret** drop-down list.
   - Select the necessary frequency of notifications in the **Monitoring notifications interval** drop-down list.

Notifications from the source about a monitoring policy triggering are repeated after the selected period until the status of the source becomes green again.

If the **Notify once** setting is selected, you receive a notification about monitoring policy activation only once.

- Turn on the **Disable monitoring notifications** toggle button if you do not want to receive notifications about the state of event sources. The toggle switch is turned off by default.

6. Click **Save**.

The SMTP server connection is now configured, and users can receive email messages from KUMA.

Working with Kaspersky Security Center tasks

You can connect Kaspersky Security Center assets to KUMA and download database and application module updates to these assets, or run an anti-virus scan on them by using Kaspersky Security Center tasks. Tasks are started in the KUMA Console.

To run Kaspersky Security Center tasks on assets connected to KUMA, it is recommended to use the following script:

1. **Creating a user account in the Kaspersky Security Center Administration Console**

   The credentials of this account are used when creating a secret to establish a connection with Kaspersky Security Center, and can be used to create a task.

   For more details about creating a user account and assigning permissions to a user, please refer to the *Kaspersky Security Center Help Guide*.

2. **Creating KUMA tasks in Kaspersky Security Center**

3. **Configuring KUMA integration with Kaspersky Security Center**

4. **Importing asset information from Kaspersky Security Center into KUMA**

   All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

   KUMA imports the following device information received from Kaspersky Security Center Network Agents:

   - Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
   - Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Click **Import**, and then select one of the following actions:
   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.
   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.
3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

   If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.
4. Click the **Import** button.

The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

5. Assigning a category to the imported assets

   After import, the assets are automatically placed in the **Uncategorized devices** group. You can assign one of the existing categories to the imported assets, or create a category and assign it to the assets.

6. **Running tasks on assets**

   You can manually start tasks in the asset information or configure tasks to start automatically.

Creating KUMA tasks in Kaspersky Security Center

You can run the anti-virus database and application module update task, and the virus scan task on Kaspersky Security Center assets connected to KUMA. The assets must have Kaspersky Endpoint Security for Windows or Kaspersky Endpoint Security for Linux installed. The tasks are created in OSMP Console.

For more details about creating the *Update* and *Virus scan* tasks on the assets with Kaspersky Endpoint Security for Windows, refer to the *Kaspersky Endpoint Security for Windows Help Guide*.

For more details about creating the *Update* and *Virus scan* tasks on the assets with Kaspersky Endpoint Security for Linux, refer to the *Kaspersky Endpoint Security for Linux Help Guide*.

Task names must begin with "kuma" (not case-sensitive and without quotations). For example, `KUMA antivirus check`. Otherwise, the task is not displayed in the list of available tasks in the KUMA Console.

Starting Kaspersky Security Center tasks manually

You can manually run the anti-virus database, application module update task, and the anti-virus scan task on Kaspersky Security Center assets connected to KUMA. The assets must have Kaspersky Endpoint Security for Windows or Kaspersky Endpoint Security for Linux installed.

First, you need to configure the integration of Kaspersky Security Center with KUMA and create tasks in Kaspersky Security Center.

To manually start a Kaspersky Security Center task:

1. In the KUMA Console, go to the **Assets** section, and select the asset that was imported from Kaspersky Security Center.

   The **Asset details** window opens.

2. Click **OSMP response**.

   This button is displayed if the connection to the Kaspersky Security Center that owns the selected asset is enabled.

3. In the opened **Select task** window, select the check boxes next to the tasks that you want to start, and click the **Start** button.

Kaspersky Security Center starts the selected tasks.

Some types of tasks are available only for certain assets.

You can obtain vulnerability and software information only for assets running a Windows operating system.

Starting Kaspersky Security Center tasks automatically

You can configure the automatic start of the anti-virus database and application module update task and the virus scan task for Kaspersky Security Center assets connected to KUMA. The assets must have Kaspersky Endpoint Security for Windows or Kaspersky Endpoint Security for Linux installed.

First, you need to configure the integration of Kaspersky Security Center with KUMA and create tasks in Kaspersky Security Center.

Configuring automatic start of Kaspersky Security Center tasks includes the following steps:

Step 1. Adding a correlation rule

To add a correlation rule:

1. In the KUMA Console, select the **Resources** section.
2. Select **Correlation rules** and click the **Add correlation rule** button.
3. On the **General** tab, define the following settings:
   a. In the **Name** field, define the rule name.
   b. In the **Tenant** drop-down list, select the tenant that owns the resource.
   c. In the **Type** drop-down list, select **simple**.
   d. In the **Propagated fields** field, add the following fields: DestinationAssetID.
   e. If required, define the values for the following fields:
      - In the **Rate limit** field, define the maximum number of times per second that the rule will be triggered.
      - In the **Severity** field, define the severity of alerts and correlation events that will be created as a result of the rule being triggered.
      - In the **Description** field, provide any additional information.
4. On the **Selectors** → **Settings** tab, do the following:
   a. In the **Filter** drop-down list, select **Create new**.
   b. In the **Conditions** field, click the **Add group** button.
   c. In the operator field for the group you added, select **AND**.
   d. Add a condition for filtering by the DeviceProduct field value:
      1. In the **Conditions** field, click the **Add condition** button.
      2. In the condition field, select **If**.
      3. In the **Left operand** field, select event field.
      4. In the **Event field** field, select DeviceProduct.
      5. In the **Operator** field, select =.
      6. In the **Right operand** field, select **constant**.
      7. In the **Value** field, enter `OSMP`.
   e. Add a condition for filtering by the Name field value:
      1. In the **Conditions** field, click the **Add condition** button.
      2. In the condition field, select **If**.

3. In the **Left operand** field, select event field.

4. In the event field, select Name.

5. In the **Operator** field, select =.

6. In the **Right operand** field, select **constant**.

7. In the **value** field, enter the name of the event. When this event is detected, the task is started automatically.

   For example, if you want the *Virus scan* task to start when Kaspersky Security Center registers the *Malicious object detected* event, specify this name in the **Value** field.

   You can view the event name in the **Name** field of the event details.

5. On the **Actions** tab, define the following settings:

   a. In the **Actions** section, open the **On every event** drop-down list.

   b. Select the **Output** check box.

   You do not need to fill in other fields.

6. Click the **Save** button.

The correlation rule will be created.

Step 2. Creating a correlator

You need to launch the correlator Installation Wizard. At step 3 of the wizard, you are required to select the correlation rule that you added by following this guide.

> The DeviceHostName field must display the domain name (FQDN) of the asset. If it is not displayed, create a DNS record for this asset and create a DNS enrichment rule at Step 4 of the wizard.

Step 3. Adding a filter

To add a filter:

1. In the KUMA Console, select the **Resources** section.

2. Select **Filters** and click the **Add filter** button.

3. In the **Name** field, specify the filter name.

4. In the **Tenant** drop-down list, select the tenant that owns the resource.

5. In the **Conditions** field, click the **Add group** button.

6. In the operator field for the group you added, select **AND**.

7. Add a condition for filtering by the DeviceProduct field value:

   a. In the **Conditions** field, click the **Add condition** button.

   b. In the condition field, select **If**.

c. In the **Left operand** field, select event field.

d. In the **Event field** field, select Type.

e. In the **Operator** field, select =.

f. In the **Right operand** field, select **constant**.

g. In the **Value** field, enter 3.

8. Add a condition for filtering by the Name field value:

a. In the **Conditions** field, click the **Add condition** button.

b. In the condition field, select **If**.

c. In the **Left operand** field, select event field.

d. In the event field, select Name.

e. In the **Operator** field, select =.

f. In the **Right operand** field, select **constant**.

g. In the **Value** field, enter the name of the correlation rule created at Step 1.

Step 4. Adding a response rule

To add a response rule:

1. In the KUMA Console, select the **Resources** section.

2. Select **Response rules** and click the **Add response rule** button.

3. In the **Name** field, define the rule name.

4. In the **Tenant** drop-down list, select the tenant that owns the resource.

5. In the **Type** drop-down list, select **Response via OSMP**.

6. In the **Open Single Management Platform task** drop-down list, select the Kaspersky Security Center task you want to start.

7. In the **Event field** drop-down list, select the DestinationAssetID.

8. In the **Workers** field, specify the number of processes that the service can run simultaneously.

   By default, the number of work processes is the same as the number of virtual processors on the server where the correlator service is installed.

- In the **Description** field, you can add up to 4,000 Unicode characters.
- In the **Filter** drop-down list, select the filter added at Step 3 of this instruction.

To send requests to Kaspersky Security Center, you must ensure that Kaspersky Security Center is available over the UDP protocol.

If a response rule is owned by the shared tenant, the displayed Kaspersky Security Center tasks that are available for selection are from the Kaspersky Security Center Server that the main tenant is connected to.

If a response rule has a selected task that is absent from the Kaspersky Security Center Server that the tenant is connected to, the task is not performed for assets of this tenant. This situation could arise when two tenants are using a common correlator, for example.

Step 5. Adding a response rule to the correlator

To add a response rule to the correlator:

1. In the KUMA Console, select the **Resources** section.
2. Select **Correlators**.
3. In the list of correlators, select the correlator added at Step 2 of this instruction.
4. In the steps tree, select **Response rules**.
5. Click **Add**.
6. In the **Response rule** drop-down list, select the rule added at step 4 of these instructions.
7. In the steps tree, select **Setup validation**.
8. Click the **Save and restart services** button.
9. Click the **Save** button.

   The response rule will be added to the correlator.

The automatic start will be configured for the anti-virus database and application module update task and the virus scan task on Kaspersky Security Center assets connected to KUMA. The tasks are started when a threat is detected on the assets and KUMA receives the corresponding events.

Checking the status of Kaspersky Security Center tasks

In the KUMA Console, you can check whether a Kaspersky Security Center task was started or whether a search for events owned by the collector listening for Kaspersky Security Center events was completed.

To check the status of Kaspersky Security Center tasks:

1. In KUMA, select **Resources** → **Active services**.

2. Select the collector that is configured to receive events from the Kaspersky Security Center Server and click the **Go to Events** button.

   A new browser tab will open in the **Events** section of KUMA. The table displays events from the Kaspersky Security Center Server. The status of the tasks can be seen in the **Name** column.

Kaspersky Security Center event fields:

- **Name**—status or type of the task.

- **Message**—message about the task or event.

- **FlexString<number>Label**—name of the attribute received from Kaspersky Security Center. For example, `FlexString1Label=TaskName`.

- **FlexString<number>**—value of the FlexString<number>Label attribute. For example, `FlexString1=Download updates`.

- **DeviceCustomNumber<number>Label**—name of the attribute related to the task state. For example, `DeviceCustomNumber1Label=TaskOldState`.

- **DeviceCustomNumber<number>**—value related to the task state. For example, `DeviceCustomNumber1=1` means the task is executing.

- **DeviceCustomString<number>Label**—name of the attribute related to the detected vulnerability: for example, a virus name, affected application.

- **DeviceCustomString<number>**—value related to the detected vulnerability. For example, the attribute-value pairs `DeviceCustomString1Label=VirusName` and `DeviceCustomString1=EICAR-Test-File` mean that the EICAR test virus was detected.

KUMA logs

Component logs

By default, only errors are logged for all KUMA components. To receive detailed data in logs, configure **Debug** mode in the component settings.

The log is appended until it reaches 5 GB. When the log reaches 5 GB, it is archived and new events are written to a new log. Archives are kept in the log folder for 7 days, after 7 days the archive is deleted. A maximum of four archived logs are stored on the server at the same time. Whenever a new log archive is created, if the total number of archives becomes greater than four, the oldest log archive is deleted.

Debug mode is available for the following components:

Services:
 	
• Storage
• Correlators
• Collectors
• Agents
	To enable it, use the **Debug** toggle switch in the settings of the service.
	
	Storage location: the service installation directory. For example, /opt/kaspersky/kuma/<service name>/log/<service name>. You can download the service logs from the KUMA Console, in the **Resources** → **Active services** section by selecting the desired service and clicking **Log**.
	
	Logs residing on Linux machines can be viewed by running the journalctl and tail command. For example:
	
	• Storage. To return the latest logs from the storage installed on the server, run the following command:
	`journalctl -f -u kuma-storage-< storage ID >`
	
	• Correlators. To return the latest logs from correlators installed on the server, run the following command:
	`journalctl -f -u kuma-correlator-< correlator ID >`
	
	• Collectors. To return the latest logs from a specific collector installed on the server, run the following command:
	`journalctl -f -u kuma-collector-< collector ID >`
	
	• Agents. To return the latest logs from an agent installed on the server, run the following command:
	`tail -f /opt/kaspersky/agent/< Agent ID >/log/agent`
	The activity of Agents on Windows machines is always logged if they are assigned the logon as a service permission. Data is specified in more detail when the **Debug** check box is selected. Agent logs on Windows machines can be viewed in the file located at the path %PROGRAMDATA%\Kaspersky Lab\KUMA\<Agent ID>\agent.log. Logs of Agents on Linux machines are stored in the agent installation directory.

Resources: • Connectors • Destinations • Enrichment rules	To enable it, use the **Debug** toggle switch in the settings of the service to which the resource is linked. The logs are stored on the machine hosting the installed service that uses the relevant resource. Detailed data for resources can be viewed in the log of the service linked to a resource.

KUMA notifications

Standard notifications

KUMA can be configured to send email notifications using an SMTP server. To do so, configure a connection to an SMTP server and select the **Receive email notifications** check box for users who should receive notifications.

KUMA automatically notifies users about the following events:

- A report was created (the users listed in the report template receive a notification).
- A task was performed (the users who created the task receive a notification).
- New resource packages are available. They can be obtained by updating the KUMA repository (the users whose email address is specified in the task settings are notified).

Editing a KUMA user profile

To edit a KUMA user profile:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Access**, click **Profile**.

   This opens the **Profile** window.

3. Specify the following settings:

   - In the **Time zone** field, select the time zone in which you want timestamps to be displayed in KUMA. The list is generated based on the settings of the user's operating system and takes into account daylight saving time. The timezone of the user's browser is used by default.

   - Select the **Use ISO 8601 format** check box to display timestamps in the ISO 8601 format (for example, 2026-02-08T17:35:27+03:00). This format is used in those sections of the KUMA web interface that use an abbreviated or full display of the date and time, and is applied throughout the KUMA system. The configured format is saved in the KUMA Core settings.

   - Select the **Receive email notifications** check box if you want the user to receive SMTP notifications from KUMA.

   - Select the **Display non-printable characters** check box if you want the KUMA web interface to display non-printing characters such as spaces, tab characters, and line breaks. If the **Display non-printable characters** check box is selected, you can press **`Ctrl/Command+*`** to enable or disable the display of non-printing characters.

     Spaces and tab characters are displayed in all input fields (except **Description**), in normalizers, correlation rules, filters, and connectors, and in SQL queries for searching events in the **Events** section. Spaces are displayed as dots. A tab character is displayed as a dash in normalizers, correlation rules, filters and connectors. In other fields, a tab character is displayed as one or two dots.

     Line break characters are displayed in all input fields that support multi-line input, such as the event search field.

   - In the **Space sets** drop-down list, select check boxes next to one or more space sets to which you want to grant access. If the user does not have a space set selected, access is granted to the default space set.

4. Click **Save**.

   The KUMA user profile is modified.

Working with tenants

Tenants are displayed in the table under Settings → KUMA → Open KUMA settings → Tenants. You can sort the table by clicking its columns.

Available columns:

- **Name** is the name of the tenant.

- **EPS limit** is the EPS (events per second) quota allocated to the tenant out of the overall EPS quota determined by the license.

- **Description** is the description of the tenant.

- **Created**is the tenant creation date.

- **Updated** is the date and time when the tenant information was most recently modified.

You can export the list of tenants, for example, to display tenant names in Active lists or Context tables widgets. In such cases, export the list of tenants to then import it to a dictionary. To do so, click the **Export** → **For dictionary** button. The file with the list of tenants is downloaded to your computer.

You can also export all information from the table of tenants to a file. To do so, click the **Export** → **All fields** button. The file with the list of tenants is downloaded to your computer.

Working with geographic data

A list of mappings of IP addresses or ranges of IP addresses to geographic data can be uploaded to KUMA for use in event enrichment.

Geodata format

Geodata can be uploaded to KUMA as a CSV file in UTF-8 encoding. A comma is used as the delimiter. The first line of the file contains the field headers: `Network,Country,Region,City,Latitude,Longitude`.

CSV file description

Field header name in CSV	Field description	Example
`Network`	IP address in one of the following formats: • Single IP address. • Range of IP addresses. • IP address in CIDR format. Mixing of IPv4 and IPv6 addresses is allowed. Required field.	• `192.168.2.24` • `192.168.2.25-192.168.2.35` • `131.10.55.70/8` • `2001:DB8::0/120`
`Country`	Country designation used by your organization. For example, this could be its name or code. Required field.	• `Russia` • `RU`
`Region`	Regional designation used by your organization. For example, this could be its name or code.	• `Sverdlovsk Oblast` • `RU-SVE`
`City`	City designation used by your organization. For example, this could be its name or code.	• `Yekaterinburg` • `65701000001`
`Latitude`	Latitude of the described location in decimal format. This field can be empty, in which case the value 0 will be used when importing data into KUMA.	`56.835556`
`Longitude`	Longitude of the described location in decimal format. This field can be empty, in which case the value 0 will be used when importing data into KUMA.	`60.612778`

Converting geographic data from MaxMind to IP2Location

KUMA can use geodata from MaxMind, IP2Location, or others in a format compatible with these. The name of the file to be converted must match the following template: (GeoIP2|GeoLite2)-(City|Country)-Blocks-IPv(4|6).csv. However, you must convert the files you want to use to a format that KUMA supports. Conversion can be done using the script below. Make sure that the files do not contain duplicate records. For example, if a file has few columns, different records may contain data from the same network with the same geodata. Such files cannot be converted. To successfully perform the conversion, make sure that there are no duplicate rows and that every row has at least one unique field.

Download script

Python 2.7 or later is required to run the script.

Script start command:

```
python converter.py --type <type of geographic data being processed: "maxmind" or "ip2location"> --out <directory where a CSV file containing geographic data in KUMA format will be placed> --input <path to the ZIP archive containing geographic data from MaxMind or IP2location>
```

When the script is run with the `--help` flag, help is displayed for the available script parameters: `python converter.py --help`

Command for converting a file containing a Russian database of IP address ranges from a MaxMind ZIP archive:

```
python converter.py --type maxmind --lang ru --input MaxMind.zip --out geoip_maxmind_ru.csv
```

If the `--lang` parameter is not specified, the script receives information from the GeoLite2-City-Locations-en.csv file from the ZIP archive by default.

Absence of the `--lang` parameter for MaxMind is equivalent to the following command:

```
python converter.py --type maxmind --input MaxMind.zip --out geoip_maxmind.csv
```

Command for converting a file from an IP2Location ZIP archive:

```
python converter.py --type ip2location --input IP2LOCATION-LITE-DB11.CSV.ZIP --out geoip_ip2location.csv
```

Command for converting a file from several IP2Location ZIP archives:

```
python converter.py --type ip2location --input IP2LOCATION-LITE-DB11.CSV.ZIP IP2LOCATION-LITE-DB11.IPV6.CSV.ZIP --out geoip_ip2location_ipv4_ipv6.csv
```

The `--lang` parameter is not used for IP2Location.

Required sets of fields

The MaxMind source files GeoLite2-City-Blocks-IPv4.csv and GeoLite2-City-Blocks-IPv6.csv must contain the following set of fields:

network,geoname_id,registered_country_geoname_id,represented_country_geoname_id,is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_radius

Example set of source data:

```
network,geoname_id,registered_country_geoname_id,represented_country_geoname_id,
is_anonymous_proxy,is_satellite_provider,postal_code,latitude,longitude,accuracy_radius

1.0.0.0/24,2077456,2077456,,0,0,,-33.4940,143.2104,1000

1.0.1.0/24,1814991,1814991,,0,0,,34.7732,113.7220,1000
```

The remaining CSV files with the locale code must contain the following set of fields:

geoname_id,locale_code,continent_code,continent_name,country_iso_code,country_name,
subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_2_name,
city_name,metro_code,time_zone,is_in_european_union

Example set of source data:

```
geoname_id,locale_code,continent_code,continent_name,country_iso_code,country_name,
subdivision_1_iso_code,subdivision_1_name,subdivision_2_iso_code,subdivision_2_name,
city_name,metro_code,time_zone,is_in_european_union

1392,de,AS,Asien,IR,Iran,02,Mazandaran,,,,,Asia/Tehran,0

7240,de,AS,Asien,IR,Iran,28,Nord-Chorasan,,,,,Asia/Tehran,0
```

The source IP2Location files must contain data on the network ranges, Country, Region, City, Latitude, and Longitude

Example set of source data:

```
"0","16777215","-","-","-","-","0.000000","0.000000","-","-"

"16777216","16777471","US","United States of America","California","Los
Angeles","34.052230","-118.243680","90001","-07:00"

"16777472","16778239","CN","China","Fujian","Fuzhou","26.061390","119.306110","350004","+0
```

If the source files contain a different set of fields than the one indicated in this section, or if some fields are missing, the missing fields in the target CSV file will be empty after conversion.

Importing and exporting geographic data

If necessary, you can manually import and export geographic data into KUMA. Geographic data is imported and exported in a CSV file. If the geographic data import is successful, the previously added data is overwritten and an audit event is generated in KUMA.

To import geographic data into KUMA:

1. Prepare a CSV file containing geographic data.

   Geographic data received from MaxMind and IP2Location must be converted to a format supported by KUMA.

2. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

3. Under **Other**, click **Common**.

   This opens the **Common** window.

4. In the **GeoIP settings** settings block, click the **Import from file** button and select a CSV file containing geographic data.

   Wait for the geographic data import to finish. The data import is interrupted if the page is refreshed.

The geographic data is uploaded to KUMA.

To export geographic data from KUMA:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Other**, click **Common**.

   This opens the **Common** window.

3. In the **GeoIP settings** settings block, click the **Export** button.

Geographic data will be downloaded as a CSV file named geoip.csv (in UTF-8 encoding) based on the settings of your browser.

The data is exported in the same format as it was uploaded, with the exception of IP address ranges. If a range of addresses was indicated in the format `1.0.0.0/24` in a file imported into KUMA, the range will be displayed in the format `1.0.0.0-1.0.0.255` in the exported file.

Default mapping of geographic data

If you select the `SourceAddress`, `DestinationAddress` and `DeviceAddress` event fields as the IP address source when configuring a geographic data enrichment rule, the **Apply default mapping** button becomes available. You can click this button to add preconfigured mapping pairs of geographic data attributes and event fields as described below.

Default mappings for the SourceAddress event field

Geodata attribute	Event field
Country	`SourceCountry`
Region	`SourceRegion`
City	`SourceCity`
Latitude	`SourceLatitude`
Longitude	`SourceLongitude`

Default mappings for the DestinationAddress event field

Geodata attribute	Event field
Country	`DestinationCountry`

Region	`DestinationRegion`
City	`DestinationCity`
Latitude	`DestinationLatitude`
Longitude	`DestinationLongitude`

Default mappings for the DeviceAddress event field

Geodata attribute	Event field
Country	`DeviceCountry`
Region	`DeviceRegion`
City	`DeviceCity`
Latitude	`DeviceLatitude`
Longitude	`DeviceLongitude`

KUMA resources

Resources are KUMA components that contain parameters for implementing various functions: for example, establishing a connection with a given web address or converting data according to certain rules. These components are assembled in various combinations into resource sets for services which are further used as the basis for creating KUMA services.

Resources are contained in the **Resources** section, **Resources** block of KUMA Console. The following resource types are available:

- Correlation rules—resources of this type contain rules for identifying event patterns that indicate threats. If the conditions specified in these resources are met, a correlation event is generated.
- Normalizers—resources of this type contain rules for converting incoming events into the format used by KUMA. After processing in the normalizer, the raw event becomes normalized and can be processed by other KUMA resources and services.
- **Connectors**—resources of this type contain settings for establishing network connections.
- **Aggregation rules**—resources of this type contain rules for combining several basic events of the same type into one aggregation event.
- **Enrichment rules**—resources of this type contain rules for supplementing events with information from third-party sources.
- Destinations—resources of this type contain settings for forwarding events to a destination for further processing or storage.
- Filters—resources of this type contain conditions for rejecting or selecting individual events from the stream of events.
- Response rules—resources of this type are used in correlators to, for example, execute scripts or launch Kaspersky Security Center tasks when certain conditions are met.
- **Notification templates**—resources of this type are used when sending notifications about new alerts.

- **Active lists**—resources of this type are used by correlators for dynamic data processing when analyzing events according to correlation rules.
- **Dictionaries**—resources of this type are used to store keys and their values, which may be required by other KUMA resources and services.
- **Proxies**—resources of this type contain settings for using proxy servers.
- **Secrets**—resources of this type are used to securely store confidential information (such as credentials) that KUMA needs to interact with external services.
- **Context tables**—resources of this type are used by KUMA correlators for analyzing events in accordance with correlation rules.

When you click on a resource type, a window opens displaying a table with the available resources of this type. The resource table contains the following columns:

- **Name**—the name of a resource. Can be used to search for resources and sort them.
- **Updated**—the date and time of the last update of a resource. Can be used to sort resources.
- **Created by**—the name of the user who created a resource.
- **Description**—the description of a resource.

The maximum table size is not limited. If you want to select all resources, scroll to the end of the table and select the **Select all** check box, which selects all available resources in the table.

Resources can be organized into folders. The folder structure is displayed in the left part of the window: root folders correspond to tenants and contain a list of all resources of the tenant. All other folders nested within the root folder display the resources of an individual folder. When a folder is selected, the resources it contains are displayed as a table in the right pane of the window.

Resources can be created, edited, copied, moved from one folder to another, and deleted. Resources can also be exported and imported.

KUMA comes with a set of predefined resources, which can be identified by the `[OOTB]< resource_name >` name. OOTB resources are protected from editing.

If you want to adapt a predefined OOTB resource to your organization's infrastructure:

1. In the **Resources**-<resource type> section, select the OOTB resource that you want to edit.
2. In the upper part of the KUMA Console, click **Duplicate**, then click **Save**.

   A new resource named `[OOTB]< resource_name > - copy` is displayed in the web interface.
3. Edit the copy of the predefined resource as necessary and save your changes.

The adapted resource is available for use.

Operations with resources

To manage KUMA resources, you can create, move, copy, edit, delete, import, and export them. These operations are available for all resources, regardless of the resource type.

KUMA resources reside in folders. You can add, rename, move, or delete resource folders.

Creating, renaming, moving, and deleting resource folders

Resources can be organized into folders. The folder structure is displayed in the left part of the window: root folders correspond to tenants and contain a list of all resources of the tenant. All other folders nested within the root folder display the resources of an individual folder. When a folder is selected, the resources it contains are displayed as a table in the right pane of the window.

You can create, rename, move and delete folders.

To create a folder:

1. Select the folder in the tree where the new folder is required.
2. Click the **Add folder** button.

   The folder will be created.

To rename a folder:

1. Locate required folder in the folder structure.
2. Hover over the name of the folder.

   The ... icon will appear near the name of the folder.
3. Open the ... drop-down list and select **Rename**.

   The folder name will become active for editing.
4. Enter the new folder name and press **ENTER**.

   > The folder name cannot be empty.

   The folder will be renamed.

To move a folder,

Drag and drop the folder to a required place in folder structure by clicking its name.

> Folders cannot be dragged from one tenant to another.

To delete a folder:

1. Locate required folder in the folder structure.
2. Hover over the name of the folder.

   The ... icon will appear near the name of the folder.

3. Open the ... drop-down list and select **Delete**.

   The conformation window appears.

4. Click **OK**.

The folder will be deleted.

> The program does not delete folders that contain files or subfolders.

Creating, duplicating, moving, editing, and deleting resources

You can create, move, copy, edit, and delete resources.

To create the resource:

1. In the main menu, go to **Monitoring** → **Resources** → **<resource type>**.

2. Select or create a folder where you want to add the new resource.

   Root folders correspond to tenants. For a resource to be available to a specific tenant, it must be created in the folder of that tenant.

3. Click the **Add <resource type>** button.

   The window for configuring the selected resource type opens. The available configuration parameters depend on the resource type.

4. Enter a unique resource name in the **Name** field.

5. Specify the required parameters (marked with a red asterisk).

6. If necessary, specify the optional parameters (not required).

7. Click **Save**.

The resource will be created and available for use in services and other resources.

To move the resource to a new folder:

1. In the main menu, go to **Monitoring** → **Resources** → **<resource type>**.

2. Find the required resource in the folder structure, and then select the check box near the resource you want to move. You can select multiple resources.

   The ⣿ icon appears near the selected resources.

3. Use the ⣿ icon to drag and drop resources to the required folder.

The resources will be moved to the new folders.

> You can only move resources to folders of the tenant in which the resources were created. Resources cannot be moved to another tenant's folders.

To copy the resource:

1. In the main menu, go to **Monitoring** → **Resources** → **<resource type>**.
2. Find the required resource in the folder structure, then select the check box next to the resource that you want to copy, and then click **Duplicate**.

   A window opens with the settings of the resource that you have selected for copying. The available configuration parameters depend on the resource type.

   The `<selected resource name> - copy` value is displayed in the **Name** field.
3. Make the necessary changes to the parameters.
4. Enter a unique name in the **Name** field.
5. Click **Save**.

The copy of the resource will be created.

To edit the resource:

1. In the main menu, go to **Monitoring** → **Resources** → **<resource type>**.
2. Find the required resource in the folder structure, and then select the resource.

   A window with the settings of the selected resource opens. The available configuration parameters depend on the resource type.
3. Make the necessary changes to the parameters.
4. Click **Save**.

The resource will be updated. If this resource is used in a service, restart the service to apply the new settings.

To delete the resource:

1. In the main menu, go to **Monitoring** → **Resources** → **<resource type>**.
2. Find the required resource in the folder structure.
3. Select the check box next to the resource that you want to delete and click **Delete**.

   A confirmation window opens.
4. Click **OK**.

The resource will be deleted.

Linking resources to services

You can now link resources (correlation rules, enrichment rules, response rules, aggregation rules, and destinations) to services such as collectors, correlators, and event routers. The available services depend on the type of the selected resource.

To link a resource to a service:

1. In the main menu, go to **Monitoring** → **Resources and services** → **<resource name>** section.

2. Select the check box next to each resource that you want to link, and in the toolbar or menu (⋮), click **Link to service** → **Link to <service name>**.

   You can also link a resource to a service when editing resource settings. To do this, click the name of a resource, select the **Services** tab, and click **Link to service** → **Link to <service name>**.

   > When managing the settings of a correlation rule, you can use the **Correlators** tab to link the rule to correlators.

3. In the displayed **<service name>** window, select the check boxes next to the services that you need, and click **OK**.

The resource is linked to the service.

To unlink a resource from a service:

1. In the main menu, go to **Monitoring** → **Resources and services** → **<resource name>** section.

2. Click the resource and select the **Services** tab.

   > For correlation rules, the **Correlators** tab is displayed.

3. Select check boxes next to each service from which you want to unlink the resource.

4. Click **Delete** and confirm the unlinking from the service.

The resource is unlinked from the service.

Updating resources

Kaspersky regularly releases packages with resources that can be imported from the repository. You can specify an email address in the settings of the **Repository update** task. After the first execution of the task, KUMA starts sending notifications about the packages available for update to the specified address. You can update the repository, analyze the contents of each update, and decide if to import and deploy the new resources in the operating infrastructure. KUMA supports updates from Kaspersky servers and from custom sources, including offline update using the update mirror mechanism. If you have other Kaspersky applications in the infrastructure, you can connect KUMA to existing update mirrors. The update subsystem expands KUMA capabilities to respond to the changes in the threat landscape and the infrastructure. The possibility of using it without direct Internet access helps ensure the privacy of the data processed by the system.

To update resources, perform the following steps:

1. Update the repository to deliver the resource packages to the repository. The repository update is available in two modes:
   - Automatic update
   - Manual update
2. Import the resource packages from the updated repository into the tenant.

For the service to start using the resources, make sure that the updated resources are mapped after performing the import. If necessary, link the resources to collectors, correlators, or agents, and update the settings.

To enable automatic update:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.
2. Under **Other**, click **Repository update**.
3. Configure the **Data refresh interval in hours**. The default value is 24 hours.
4. Specify the **Update source**. The following options are available:
   - Kaspersky update servers.

     You can view the list of servers in the Knowledge Base.
   - Custom source:
     - The URL to the shared folder on the HTTP server.
     - The full path to the local folder on the host where the KUMA Core is installed.

       If a local folder is used, the kuma system user must have read access to this folder and its contents.
5. Specify the **Emails for notification** by clicking the **Add** button. The notifications that new packages or new versions of the packages imported into the tenant are available in the repository are sent to the specified email addresses.

   If you specify the email address of a KUMA user, the **Receive email notifications** check box must be selected in the user profile. For emails that do not belong to any KUMA user, the messages are received without additional settings. The settings for connecting to the SMTP server must be specified in all cases.
6. Click **Save**. The update task starts shortly. Then the task restarts according to the schedule.

To manually start the repository update:

1. To disable automatic updates, in the **Settings** → **KUMA** → **Open KUMA settings** → **Repository update** section, select the **Disable automatic update** check box. This check box is cleared by default. You can also start a manual repository update without disabling automatic update. Starting an update manually does not affect the automatic update schedule.

2. Specify the **Update source**. The following options are available:

   - Kaspersky update servers.

   - Custom source:

     - The URL to the shared folder on the HTTP server.

     - The full path to the local folder on the host where the KUMA Core is installed.

       If a local folder is used, the kuma user must have access to this folder and its contents.

3. Specify the **Emails for notification** by clicking the **Add** button. The notifications that new packages or new versions of the packages imported into the tenant are available in the repository are sent to the specified email addresses.

   If you specify the email address of a KUMA user, the **Receive email notifications** check box must be selected in the user profile. For emails that do not belong to any KUMA user, the messages are received without additional settings. The settings for connecting to the SMTP server must be specified in all cases.

4. Click **Run update**. Thus, you simultaneously save the settings and manually start the **Repository update** task.

Configuring a custom source using Kaspersky Update Utility

You can update resources without Internet access by using a custom update source via the Kaspersky Update Utility.

Configuration consists of the following steps:

1. Configuring a custom source using Kaspersky Update Utility:

   a. Installing and configuring Kaspersky Update Utility on one of the computers in the corporate LAN.

   b. Configuring copying of updates to a shared folder in Kaspersky Update Utility settings.

2. Configuring update of the KUMA repository from a custom source.

Configuring a custom source using Kaspersky Update Utility:

You can download the Kaspersky Update Utility distribution kit from the Kaspersky Technical Support website.

1. In Kaspersky Update Utility, enable the download of updates for KUMA:

   - Under **Applications** – **Perimeter control**, select the check box next to KUMA to enable the update capability.

   - If you work with Kaspersky Update Utility using the command line, add the following line to the [ComponentSettings] section of the updater.ini configuration file or specify the `true` value for an existing line:

     ```
     KasperskyUnifiedMonitoringAndAnalysisPlatform_XDR_1_1=true
     ```

2. In the **Downloads** section, specify the update source. By default, Kaspersky update servers are used as the update source.

3. In the **Downloads** section, in the **Update folders** group of settings, specify the shared folder for Kaspersky Update Utility to download updates to. The following options are available:

   - Specify the local folder on the host where Kaspersky Update Utility is installed. Deploy the HTTP server for distributing updates and publish the local folder on it. In OSMP Console, in the **Settings** → **KUMA** → **Open KUMA settings** → **Repository update** → **Custom source** section, specify the URL of the local folder published on the HTTP server.

   - Specify the local folder on the host where Kaspersky Update Utility is installed. Make this local folder available over the network. Mount the network-accessible local folder on the host where KUMA is installed. In OSMP Console, in the **Settings** → **KUMA** → **Open KUMA settings** → **Repository update** → **Custom source** section, specify the full path to the local folder.

For detailed information about working with Kaspersky Update Utility, refer to the Kaspersky Knowledge Base.

Exporting resources

> If shared resources are hidden for a user, the user cannot export shared resources or resources that use shared resources.

To export resources:

1. In the main menu, go to **Monitoring** → **Resources** → **Correlation rules**.

2. Click **Export resources**.

   The **Export resources** window opens with the tree of all available resources.

3. In the **Password** field enter the password that must be used to protect exported data.

4. In the **Tenant** drop-down list, select the tenant whose resources you want to export.

5. Check boxes near the resources you want to export.

   If selected resources are linked to other resources, linked resources will be exported, too.

6. Click the **Export** button.

The resources in a password-protected file are saved on your computer using your browser settings. The Secret resources are exported blank.

Importing resources

To import resources:

1. In the main menu, go to **Monitoring** → **Resources**.

2. Click **Import resources**.

   The **Resource import** window opens.

3. In the **Tenant** drop-down list, select the tenant to assign the imported resources to.

4. In the **Import source** drop-down list, select one of the following options:

   - **File**

     If you select this option, enter the password and click the **Import** button.

   - **Repository**

     If you select this option, a list of packages available for import is displayed. We recommend you to ensure that the repository update date is relatively recent and configure automatic updates if necessary.

     You can select one or more packages to import and click the **Import** button. The dependent resources of the Shared tenant are imported into the Shared tenant, the rest of the resources are imported into the selected tenant. You do not need special rights for the Shared tenant; you must only have the right to import in the selected tenant.

     The imported resources can only be deleted. To rename, edit or move an imported resource, make a copy of the resource by clicking the **Duplicate** button and perform the desired actions with the resource copy. When importing future versions of the package, the duplicate is not updated because it is a separate object.

5. Resolve the conflicts between the resources imported from the file and the existing resources if they occur. Read more about resource conflicts below.

   a. If the name, type, and guid of an imported resource fully match the name, type, and guid of an existing resource, the **Conflicts** window opens with the table displaying the type and the name of the conflicting resources. Resolve displayed conflicts:

      - To replace the existing resource with a new one, click **Replace**.

        To replace all conflicting resources, click **Replace all**.

      - To leave the existing resource, click **Skip**.

        For dependent resources, that is, resources that are associated with other resources, the **Skip** option is not available; you can only **Replace** dependent resources.

        To keep all existing resources, click **Skip all**.

   b. Click the **Resolve** button.

The resources are imported to KUMA. The Secret resources are imported blank.

Importing resources that use the extended event schema

If you import a normalizer that uses one or more fields of the extended event schema, KUMA automatically creates an extended schema field that is used in the normalizer.

If you import other types of resources that use fields of the extended event schema in their logic, the resources are imported successfully. To ensure the functioning of imported resources, you must create the corresponding fields of the extended event schema in a resource of the "normalizer" type.

If a normalizer that uses an extended event schema field is imported into KUMA and the same field already exists in KUMA, the previously created field is used.

About conflict resolving

When resources are imported into KUMA from a file, they are compared with existing resources; the following parameters are compared:

- Name and kind. If an imported resource's name and kind parameters match those of the existing one, the imported resource's name is automatically changed.
- ID. If identifiers of two resources match, a conflict appears that must be resolved by the user. This could happen when you import resources to the same KUMA server from which they were exported.

When resolving a conflict you can choose either to *replace existing resource* with the imported one or to *keep exiting resource*, skipping the imported one.

Some resources are linked: for example, in some types of connectors, the connector secret must be specified. The secrets are also imported if they are linked to a connector. Such linked resources are exported and imported together.

Special considerations of import:

1. Resources are imported to the selected tenant.
2. If a linked resource was in the Shared tenant, it ends up in the Shared tenant when imported.
3. In the **Conflicts** window, the **Parent** column always displays the top-most parent resource among those that were selected during import.
4. If a conflict occurs during import and you choose to replace existing resource with a new one, it would mean that all the other resources linked to the one being replaced are automatically replaced with the imported resources.

Known errors:

1. The linked resource ends up in the tenant specified during the import, and not in the Shared tenant, as indicated in the **Conflicts** window, under the following conditions:

   a. The associated resource is initially in the Shared tenant.

   b. In the **Conflicts** window, you select **Skip** for all parent objects of the linked resource from the Shared tenant.

   c. You leave the linked resource from the Shared tenant for replacement.

2. After importing, the categories do not have a tenant specified in the filter under the following conditions:

   a. The filter contains linked asset categories from different tenants.

   b. Asset category names are the same.

   c. You are importing this filter with linked asset categories to a new server.

3. In Tenant 1, the name of the asset category is duplicated under the following conditions:

   a. in Tenant 1, you have a filter with linked asset categories from Tenant 1 and the Shared tenant.

   b. The names of the linked asset categories are the same.

   c. You are importing such a filter from Tenant 1 to the Shared tenant.

4. You cannot import conflicting resources into the same tenant.

   The error *Unable to import conflicting resources into the same tenant* means that the imported package contains conflicting resources from different tenants and cannot be imported into the Shared tenant.

   Solution: Select a tenant other than Shared to import the package. In this case, during the import, resources originally located in the Shared tenant are imported into the Shared tenant, and resources from the other tenant are imported into the tenant selected during import.

5. Only the Main Administrator can import categories into the Shared tenant.

   The error *Only the Main administrator can import categories into the Shared tenant* means that the imported package contains resources with linked shared asset categories. You can see the categories or resources with linked shared asset categories in the KUMA Core log. Path to the Core log:

   /opt/kaspersky/kuma/core/log/core

   Solution. Choose one of the following options:

   - Do not import resources to which shared categories are linked: clear the check boxes next to the relevant resources.

   - Perform the import under a Main administrator account.

6. Only the Main administrator can import resources into the Shared tenant.

   The error *Only the Main administrator can import resources into the Shared tenant* means that the imported package contains resources with linked shared resources. You can see the resources with linked shared resources in the KUMA Core log. Path to the Core log:

   /opt/kaspersky/kuma/core/log/core

Solution. Choose one of the following options:

- Do not import resources that have linked resources from the Shared tenant, and the shared resources themselves: clear the check boxes next to the relevant resources.
- Perform the import under a Main administrator account.

Tag management

To help manage resources, KUMA lets you add tags to resources. You can use tags to search for a component, as well as manage tags, link or unlink tags.

> You cannot add tags to resources that are created from the interface of other resources. Tags can be added only from the resource's own card. You also cannot add tags to a resource that is not editable.

Tag management

The list of tags is displayed in the **Settings** → **KUMA** → **Open KUMA settings** → **Tags** section and is displayed as a table with the following columns: **Name**, **Tenant**, **Used in resources**.

In the **Tags** table, you can:

- Sort tags by **Name**, **Used in resources** fields.
- Filter by values of the **Tenant** field.
- Find a tag by the **Name** field.
- Go to the list of resources that have the selected tag.

Adding a tag

To add a tag:

1. In the main menu, go to **Monitoring** → **Resources and services**, and then select a resource.
2. In the panel above the table, click **Add**.
3. In the **Tags** field of the selected resource, add a new tag, or select a tag from the list.
4. Click **Create**.

   The new tag is added.

You can also add tags from existing ones.

When adding a tag, keep in mind the following special considerations:

- You can add multiple tags.
- A tag can contain characters of various alphabets (for example, Cyrillic, Latin, or Greek characters), numerals, underscores, and spaces.
- A tag may not contain any special characters other than the underscore and the space.
- You can enter the tag in uppercase or lowercase, but after saving, the tag is always displayed in lowercase.
- The tag inherits the tenant of the resource in which it is used.
- A tag is part of a resource and exists as long as the resource exists in which the tag was created or is used.
- Tags are unique within a tenant.
- Tags are imported or exported together with the resource as part of the resource.

Searching by tags

In the **Resources** section, you can search for resources:

- By tags
- By resource name

The search is performed across all resource types and services.

The search results display a list of resources and services.

To find resources by tags:

1. In the main menu, go to **Monitoring** → **Resources and services**, and then select a resource.
2. In the table of the resource, select the **Tags** column.
3. In the **Search** field that is displayed, enter or select a tag name.

   Searching by tag in the resource itself and in the list of resources returns full as well as partial matches of the entered text with the name of the tag.

Displays a list of resources if the specified tag is used in those resources.

In the list of resources, you can:

- Sort the list by name and type of resource or service.
- Filter resources or services by resource or service type, or by tag.
- Link or unlink tags.

Displaying tags while adding dependent resources

While adding a dependent resource, the tags of this resource are displayed in a drop-down list under **Filter settings**.

Resource tags that to not fit a single line are hidden. A special indicator lets you see that hidden tags exist.

The drop-down list can be filtered by resource name and by resource tags.

Linking and unlinking tags

To link tags to a resource or unlink tags from a resource:

1. In the main menu, go to **Monitoring** → **Resources and services**.
2. Select the **List** tab.
3. In the **Name** column, select the check boxes next to the relevant resources.
4. In the panel above the list, select the **Tags** tab.
5. Click **Link** or **Unlink**, select the tags that you need or create a new tag.

   The selected tags are linked to or unlinked from the resources.

   You can also link or unlink tags in the card of the resource.

Destinations

Destinations define network settings for sending normalized events. Collectors and correlators use destinations to describe where to send processed events. Typically, the destination points are the correlator and storage.

The settings of destinations are configured on two tabs: **Basic settings** and **Advanced settings**. The available settings depend on the selected type of destination:

- **internal**—used for receiving data from KUMA services using the 'internal' protocol.
- **nats-jetstream**—used for NATS communications.
- **tcp**—used for communications over TCP.
- **udp** – used for communication over UDP.
- **http**—used for HTTP communications.
- **diode**—used to transmit events using a data diode.
- **kafka**—used for Kafka communications.
- **file**—used for writing to a file.
- **storage**—used to transmit data to the storage.

- correlator—used to transmit data to the correlator.
- eventRouter—used for sending events to an event router.

Destination, internal type

Destinations of the **internal** typeUsed for receiving data from KUMA services using the 'internal' protocol. You can send the following data over the 'internal' protocol:

- Internal data, such as event routes.
- File attributes. If while creating the collector at the Transport step of the Installation Wizard, you specified a connector of the file, 1c-xml, or 1c-log type, at the Event parsing step, in the **Mapping** table, you can pass the name of the file being processed by the collector or the path to the file in the KUMA event field. To do this, in the **Source** column, specify one of the following values:
  - `$kuma_fileSourceName` to pass the name of the file being processed by the collector in the KUMA event field.
  - `$kuma_fileSourcePath` to pass the path to the file being processed by the collector in the KUMA event field.

  When you use a **file**, **1c-xml**, or **1c-log** connector, the new variables in the normalizer will only work with **destinations of the internal type**.
- Events to the event router. The event router can only receive events over the 'internal' protocol, therefore you can only use **internal** destinations when sending events to the event router.

Settings for a destination of the **internal** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
State	This toggle switch enables sending events to the destination. This toggle switch is turned on by default.
Type	Destination type: **internal**. Required setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.

Setting	Description
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 1,048,576 bytes (1 MB). Maximum buffer size: 67,108,864 bytes (64 MB).
Buffer flush interval	Interval (in seconds) for sending events to the destination. The default value is 1 second.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. To determine the number of handlers, you can use the following formula: (<number of CPUs> / 2) + 2. The value must be a positive integer.
Proxy server	The proxy server for the destination. You can select an existing proxy server or create a new proxy server. To create a new proxy server, select **Create new**. If you want to edit the settings of an existing proxy server, click the pencil icon next to it.
URL selection policy	Method of determining URLs to which events must be sent first if you added multiple URLs in the **URL** field on the **Basic settings**: • **Any** means events are sent to a randomly selected available URL as long as the URL accepts events. If the URL becomes unavailable, events are sent to another randomly selected available URL. This value is selected by default. • **Prefer first** means events are sent to the first added URL. If the URL becomes unavailable, events are sent to the next added available URL. If the first added URL becomes available again, events are sent to the first added URL again. • **Round robin** means events are evenly balanced among the available URLs. This method does not guarantee that events are evenly balanced among the URLs because the buffer may overflow or events may be sent to the destination. You can specify the buffer size in bytes in the **Buffer size limit** field; you can also specify the interval in seconds for sending events to the destination in the **Buffer flush interval** field.
Health check timeout	Interval, in seconds, for checking the health of the destination.
Disk buffer disabled	This toggle switch that enables the disk buffer. This toggle switch is turned on by default. The disk buffer is used if the collector cannot send normalized events to the destination. You can specify the size of the disk buffer in the **Disk buffer size limit** field. If the disk buffer runs out of free space, new normalized events will overwrite old normalized events, starting with the oldest.
Timeout	The time, in seconds, for which the destination waits for a response from another service or component.
Debug	This switch enables resource logging. The toggle switch is turned off by default.

Setting	Description
Filter	Filter for determining which events must be processed by the resource. You can select an existing filter or create a new filter. To create a new filter, select **Create new**. If you want to edit the settings of an existing filter, click the pencil icon next to it. **How to create a filter?** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

nats-jetstream type

The **nats-jetstream** type is used for NATS communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **nats-jetstream**.
URL	Required setting. URL that you want to connect to.
Topic	Required setting. The topic of NATS messages. Must contain Unicode characters.
Delimiter	Specify a character that defines where one event ends and the other begins. By default, `\n` is used.

Setting	Description
Authorization	Type of authorization when connecting to the specified URL. Possible values: • **disabled** is the default value. • **plain** — if this option is selected, you must indicate the secret containing user account credentials for authorization when connecting to the connector. **Add secret** *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Cluster ID	ID of the NATS cluster.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF If the CEF format is selected, the sent event contains the CEF header and only non-empty fields.

Setting	Description

TLS mode	Use of TLS encryption. Available values:
	• **Disabled** (default) means TLS encryption is not used.
	• **Enabled** means encryption is used, but the certificate is not verified.
	• **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
	• **Custom CA** means encryption is used with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.
	Creating a certificate signed by a Certificate Authority
	You can create a CA-signed certificate on the KUMA Core server (the following command examples use OpenSSL).
	To create a certificate signed by a Certificate Authority:
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt`
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=< common host name of KUMA server >" -out server.csr`
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	5. Upload the created `server.crt` certificate in the KUMA Console to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	When using TLS, it is impossible to specify an IP address as a URL.

Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Tcp type

The **tcp** type is used for TCP communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **tcp**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported. When using IPv6 addresses, you must also specify the interface in the `[address%interface]:port` format. For example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`).
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Timeout	The time (in seconds) to wait for a response from another service or component. The default value is `30`.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF If the CEF format is selected, the sent event contains the CEF header and only non-empty fields.
TLS mode	TLS encryption mode using certificates in pem x509 format. Available values: • **Disabled** means TLS encryption is not used. The default value. • **Enabled** means encryption is used, but certificates are not verified. • **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during application installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/. When using TLS, it is impossible to specify an IP address as a URL.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In this section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Udp type

Destinations of the **udp** type are used for UDP communications. Settings for a destination of the **udp** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
State	This toggle switch enables sending events to the destination. This toggle switch is turned on by default.
Type	Destination type, **udp**. Required setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. You cannot save a resource or service if the **URL** field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 67,108,864 (64 MB), displayed as 0 in the interface.
Buffer flush interval	Interval (in seconds) for sending events to the destination. The default value is 1 second.
Disk buffer size limit	Size of the disk buffer in bytes. Default value: 10 GB, displayed as 0 in the interface.
Handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999.
Output format	Format in which events are sent to the destination: • **JSON**. • **CEF**. If this value is selected, the transmitted events contain the CEF header and only non-empty fields. • **Syslog-CEF**

Setting	Description
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during application installation and are stored on the KUMA Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated.
Compression	Drop-down list for configuring Snappy compression: • **Disabled**. This value is selected by default. • **Use Snappy**.
URL selection policy	Method of determining URLs to which events must be sent first if you added multiple URLs in the **URL** field on the **Basic settings** tab: • **Any** means events are sent to a randomly selected available URL as long as the URL accepts events. If the URL becomes unavailable, events are sent to another randomly selected available URL. This value is selected by default. • **Prefer first** means events are sent to the first added URL. If the URL becomes unavailable, events are sent to the next added available URL. If the first added URL becomes available again, events are sent to the first added URL again. • **Round robin** means events are evenly balanced among the available URLs. This method does not guarantee that events are evenly balanced among the URLs because the buffer may overflow or events may be sent to the destination. You can specify the buffer size in bytes in the **Buffer size limit** field; you can also specify the interval in seconds for sending events to the destination in the **Buffer flush interval** field.
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged.
Disk buffer disabled	This toggle switch that enables the disk buffer. This toggle switch is turned on by default. The disk buffer is used if the collector cannot send normalized events to the destination. You can specify the size of the disk buffer in the **Disk buffer size limit** field. If the disk buffer runs out of free space, new normalized events will overwrite old normalized events, starting with the oldest.
Timeout	The time, in seconds, for which the destination waits for a response from another service or component.
Debug	This toggle switch enables logging for the resource. This toggle switch is disabled by default.

Setting	Description
Filter	Filter for determining which events must be processed by the resource. You can select an existing filter or create a new filter. To create a new filter, select **Create new**. If you want to edit the settings of an existing filter, click the pencil icon next to it. **How to create a filter?** All fields of a KUMA event have a default value. For all numeric fields, the default value is 0. For all string fields, the default value is an empty string. An event not having a field means the field has the default value. Therefore, an event can trigger a filter even if the event does not have the field specified in the filter. For example, if you specify a `DestinationPort <= 1024` condition in the filter, but the event does not contain the `DestinationPort` field, the filter is triggered anyway because `DestinationPort = 0` by default. *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. Under **Conditions**, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, you may see fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **Operator** drop-down list, select an operator. **Filter operators** • **=** means the left operand equals the right operand. • **<** means the left operand is less than the right operand. • **<=** means the left operand is less than or equal to the right operand. • **>** means the left operand is greater than the right operand. • **>=** means the left operand is greater than or equal to the right operand. • **inSubnet** means the left operand (IP address) is in the subnet of the right operand (subnet). • **contains** means the left operand contains values of the right operand. • **startsWith** means the left operand starts with one of the values of the right operand. • **endsWith** means the left operand ends with one of the values of the right operand. • **match** means the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit** checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability** checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • The **inActiveList** operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary** checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory** means the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup** means the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • The **TIDetect** operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

Setting	Description
	• **inContextTable** checks for the presence of the entry in the specified context table. • **intersect** checks for the presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. This check box does not apply to the **InSubnet**, **InActiveList**, **InCategory**, or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, from the **If** drop-down list, select **If not**. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Http type

The **http** type is used for HTTP communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **http**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`).

Setting	Description
Authorization	Type of authorization when connecting to the specified URL Possible values: • **disabled** is the default value. • **plain:** if this option is selected, you must indicate the secret containing user account credentials for authorization when connecting to the connector. **Add secret** *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Timeout	The time (in seconds) to wait for a response from another service or component. The default value is `30`.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF If the CEF format is selected, the sent event contains the CEF header and only non-empty fields.

Setting	Description

TLS mode	Use of TLS encryption. Available values:
	• **Disabled** (default) means TLS encryption is not used.
	• **Enabled** means encryption is used, but the certificate is not verified.
	• **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
	• **Custom CA** means encryption is used with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.
	Creating a certificate signed by a Certificate Authority
	You can create a CA-signed certificate on the KUMA Core server (the following command examples use OpenSSL).
	To create a certificate signed by a Certificate Authority:
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt`
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=< common host name of KUMA server >" -out server.csr`
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	5. Upload the created `server.crt` certificate in the KUMA Console to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	When using TLS, it is impossible to specify an IP address as a URL.

URL selection policy	From the drop-down list, you can select the method of deciding which URL to send events to if multiple URLs are specified. Available values: • **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination. • **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again. • **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. \n is used by default.
Path	The path that must be added for the URL request. For example, if you specify the path `/input` and enter `10.10.10.10` for the URL, requests for `10.10.10.10/input` will be sent from the destination.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	The number of services that are processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Health check path	The URL for sending requests to obtain health information about the system that the destination resource is connecting to.
Health check timeout	Frequency of the health check in seconds.
Health Check Disabled	Check box that disables the health check.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.

Setting	Description
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Diode type

The **diode** type is used to transmit events using a data diode.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **diode**.

Setting	Description
Data diode source directory	Required setting. The directory from which the data diode moves events. The path can contain up to 255 Unicode characters. **Limitations when using prefixes in paths on Windows servers** On Windows servers, absolute paths to directories must be specified. Directories with names matching the following regular expressions cannot be used: • `^[a-zA-Z]:\\Program Files` • `^[a-zA-Z]:\\Program Files \(x86\)` • `^[a-zA-Z]:\\Windows` • `^[a-zA-Z]:\\ProgramData\\Kaspersky Lab\\Open Single Management Platform` **Limitations when using prefixes in paths on Linux servers** Prefixes that cannot be used when specifying paths to files: • `/*` • `/bin` • `/boot` • `/dev` • `/etc` • `/home` • `/lib` • `/lib64` • `/proc` • `/root` • `/run` • `/sys` • `/tmp` • `/usr/*` • `/usr/bin/` • `/usr/local/*` • `/usr/local/sbin/` • `/usr/local/bin/` • `/usr/sbin/` • `/usr/lib/` • `/usr/lib64/` • `/var/*` • `/var/lib/` • `/var/run/` • `/opt/kaspersky/kuma/`

Setting	Description
	Files are available at the following paths: • `/opt/kaspersky/kuma/clickhouse/logs/` • `/opt/kaspersky/kuma/mongodb/log/` • `/opt/kaspersky/kuma/victoria-metrics/log/`
Temporary directory	Directory in which events are prepared for transmission to the data diode. Events are stored in a file when a timeout (10 seconds by default) or a buffer overflow occurs. The prepared file is moved to the directory specified in the **Data diode source directory** field. The checksum (SHA256) of the file contents is used as the name of the file containing events. The temporary directory must be different from the data diode source directory.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Compression	You can use Snappy compression. By default, compression is **disabled**. This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used. This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.

Setting	Description
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Kafka type

The **kafka** type is used for Kafka communications.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **kafka**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`). You can add multiple addresses by clicking the URL button.
Topic	Required setting. Subject of Kafka messages. Must contain from 1 to 255 of the following characters: a–z, A–Z, 0–9, ".", "_", "-".
Delimiter	Specify a character that defines where one event ends and the other begins. By default, `\n` is used.

Setting	Description
Authorization	Type of authorization when connecting to the specified URL Possible values: • **disabled** is the default value. • **PFX** — a certificate must be generated with a private key in PKCS#12 container format in an external Certificate Authority. Then the certificate must be exported from the key store and uploaded to the KUMA Console as a PFX secret. • **Add PFX secret** *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX certificate with the private key. 3. In the **Password** field, enter the PFX certificate security password that was set in the PFX Certificate Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list. • **plain** — you must indicate the secret containing user account credentials for authorization when connecting to the connector. **Add secret** *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Timeout	The time (in seconds) to wait for a response from another service or component. The default value is `30`.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF If the CEF format is selected, the sent event contains the CEF header and only non-empty fields.

Setting	Description

TLS mode	Use of TLS encryption. Available values:
	• **Disabled** (default) means TLS encryption is not used.
	• **Enabled**—use encryption without certificate verification.
	• **With verification** means encryption is used with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
	• **Custom CA** means encryption is used with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.
	Creating a certificate signed by a Certificate Authority
	You can create a CA-signed certificate on the KUMA Core server (the following command examples use OpenSSL).
	To create a certificate signed by a Certificate Authority:
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt`
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=< common host name of KUMA server >" -out server.csr`
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	5. Upload the created `server.crt` certificate in the KUMA Console to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	When using TLS, it is impossible to specify an IP address as a URL.

Delimiter	In the drop-down list, you can select the character to mark the boundary between events. By default, `\n` is used.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In this section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

File type

The **file** type is used for writing data to a file.

> If you delete a destination of the 'file' type used in a service, that service must be restarted.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **file**.

Setting	Description
URL	Required setting. Path to the file to which the events must be written. **Limitations when using prefixes in file paths** Prefixes that cannot be used when specifying paths to files: • `/*` • `/bin` • `/boot` • `/dev` • `/etc` • `/home` • `/lib` • `/lib64` • `/proc` • `/root` • `/run` • `/sys` • `/tmp` • `/usr/*` • `/usr/bin/` • `/usr/local/*` • `/usr/local/sbin/` • `/usr/local/bin/` • `/usr/sbin/` • `/usr/lib/` • `/usr/lib64/` • `/var/*` • `/var/lib/` • `/var/run/` • `/opt/kaspersky/kuma/` Files are available at the following paths: • `/opt/kaspersky/kuma/clickhouse/logs/` • `/opt/kaspersky/kuma/mongodb/log/` • `/opt/kaspersky/kuma/victoria-metrics/log/`
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Delimiter	In the drop-down list, you can select the character to mark the boundary between events. \n is used by default.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Output format	Format for sending events to an external destination. Available values: • JSON • CEF If the CEF format is selected, the sent event contains the CEF header and only non-empty fields.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Storage type

The **storage** type is used to transmit data to the storage.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **storage**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`). You can add multiple addresses by clicking the **URL** button. The **URL** field supports search for services by FQDN, IP address, and name. Search string formats: • `<Search value>` —search is performed by FQDN, IP addresses, and service names. • `<First search value ending in one or more digits>:<second search value>` —the first value is used to search by the service FQDN or IP address, and the second value is used to search by port. • `:<value>` —search is performed by port.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Proxy server	Drop-down list for selecting a proxy server.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
URL selection policy	Drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified: • **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination. • **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again. • **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Health check timeout	Frequency of the health check in seconds.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In this section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Correlator type

The **correlator** type is used to transmit data to the correlator.

Basic settings tab

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
The **State** toggle switch	Used when events must be sent to the destination. By default, sending events is enabled.
Type	Required setting. Destination type, **correlator**.
URL	Required setting. URL that you want to connect to. Available formats: `host:port`, `IPv4:port`, `:port`. IPv6 addresses are also supported, however, when you use them, you must specify the interface as well: `[address%interface]:port`. Example: `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`). You can add multiple addresses by clicking the **URL** button. The **URL** field supports search for services by FQDN, IP address, and name. Search string formats: • `<Search value>` —search is performed by FQDN, IP addresses, and service names. • `<First search value ending in one or more digits>:<second search value>` —the first value is used to search by the service FQDN or IP address, and the second value is used to search by port. • `:<value>` —search is performed by port.
Description	Resource description: up to 4,000 Unicode characters.

Advanced settings tab

Setting	Description
Proxy server	Drop-down list for selecting a proxy server.
Buffer size	Sets the size of the buffer. The default value is 1 KB, and the maximum value is 64 MB.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
URL selection policy	Drop-down list in which you can select a method for determining which URL to send events to if several URLs have been specified: • **Any**. Events are sent to one of the available URLs as long as this URL receives events. If the connection is broken (for example, the receiving node is disconnected) a different URL will be selected as the events destination. • **Prefer first**. Events are sent to the first URL in the list of added addresses. If it becomes unavailable, events are sent to the next available node in sequence. When the first URL becomes available again, events start to be sent to it again. • **Balanced** means that packages with events are evenly distributed among the available URLs from the list. Because packets are sent either on a destination buffer overflow or on the flush timer, this URL selection policy does not guarantee an equal distribution of events to destinations.
Buffer flush interval	Time (in seconds) between sending batches of data to the destination. The default value is `1 second`.
Number of handlers	This field is used to set the number of services processing the queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.
Health check timeout	Frequency of the health check in seconds.
Debug	This toggle switch lets you specify whether resource logging must be enabled. The default value is **Disabled**.
Disk buffer disabled	Drop-down list that lets you enable or disable the disk buffer. By default, the disk buffer is enabled. The disk buffer is used if the collector cannot send normalized events to the destination. The amount of allocated disk space is limited by the value of the **Disk buffer size limit** setting. If the disk space allocated for the disk buffer is exhausted, events are rotated as follows: new events replace the oldest events written to the buffer.

Setting	Description
Filter	In the **Filter** section, you can specify the criteria for identifying events that must be processed by the resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Destination, eventRouter type

Destinations of the **eventRouter** typeUsed for sending events to an event router. Settings for a destination of the **eventRouter** type are described in the following tables.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
State	This toggle switch enables sending events to the destination. This toggle switch is turned on by default.
Type	Destination type: **eventRouter**. Required setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.
Tags	Tags for resource search. Optional setting.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Buffer size	Buffer size in bytes for accumulating events in the RAM of the server before sending them for further processing or storage. The value must be a positive integer. Default buffer size: 1,048,576 bytes (1 MB). Maximum buffer size: 67,108,864 bytes (64 MB).
Buffer flush interval	Interval (in seconds) for sending events to the destination. The default value is 1 second.
Disk buffer size limit	Size of the disk buffer in bytes. The default value is 10 GB.
Handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. To determine the number of handlers, you can use the following formula: (<number of CPUs> / 2) + 2. The value must be a positive integer.
Proxy server	The proxy server for the destination. You can select an existing proxy server or create a new proxy server. To create a new proxy server, select **Create new**. If you want to edit the settings of an existing proxy server, click the pencil icon next to it.
URL selection policy	Method of determining URLs to which events must be sent first if you added multiple URLs in the **URL** field on the **Basic settings**: • **Any** means events are sent to a randomly selected available URL as long as the URL accepts events. If the URL becomes unavailable, events are sent to another randomly selected available URL. This value is selected by default. • **Prefer first** means events are sent to the first added URL. If the URL becomes unavailable, events are sent to the next added available URL. If the first added URL becomes available again, events are sent to the first added URL again. • **Round robin** means events are evenly balanced among the available URLs. This method does not guarantee that events are evenly balanced among the URLs because the buffer may overflow or events may be sent to the destination. You can specify the buffer size in bytes in the **Buffer size limit** field; you can also specify the interval in seconds for sending events to the destination in the **Buffer flush interval** field.
Health check timeout	Interval, in seconds, for checking the health of the destination.
Disk buffer disabled	This toggle switch that enables the disk buffer. This toggle switch is turned on by default. The disk buffer is used if the collector cannot send normalized events to the destination. You can specify the size of the disk buffer in the **Disk buffer size limit** field. If the disk buffer runs out of free space, new normalized events will overwrite old normalized events, starting with the oldest.
Timeout	The time, in seconds, for which the destination waits for a response from another service or component.
Debug	This switch enables resource logging. The toggle switch is turned off by default.

Setting	Description
Filter	Filter for determining which events must be processed by the resource. You can select an existing filter or create a new filter. To create a new filter, select **Create new**. If you want to edit the settings of an existing filter, click the pencil icon next to it. **How to create a filter?** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when managing the settings of a destination and is used for linking to collectors, correlators, and event routers.

To link a destination to a service, click the **Link to service** button → **Link to <service_name>**. This opens a window; in that window, select one or more services and click **OK**. The destination is linked to the selected services.

If necessary, you can unlink the destination from a service. To do this, select the check box next to the service, click **Delete** and confirm the unlinking from the service.

Predefined destinations

Destinations listed in the table below are included in the OSMP distribution kit.

Predefined destinations

Destination name	Description
[OOTB] Correlator	Sends events to a correlator.
[OOTB] Storage	Sends events to storage.

Normalizers

Normalizers are used for converting raw events that come from various sources in different formats to the KUMA event data model. Normalized events become available for processing by other KUMA resources and services.

A normalizer consists of the *main* event parsing rule and optional *additional event parsing rules*. By creating a main parsing rule and a set of additional parsing rules, you can implement complex event processing logic. Data is passed along the tree of parsing rules depending on the conditions specified in the **Extra normalization conditions** setting. The sequence in which parsing rules are created is significant: the event is processed sequentially and the processing sequence is indicated by arrows.

The following event normalization options are now available:

- 1 collector — 1 normalizer

  We recommend using this method if you have many events of the same type or many IP addresses from which events of the same type may originate. You can configure one collector with only one normalizer, which is optimal in terms of performance.

- 1 collector — multiple normalizers linked to IP

  This method is available for collectors with a connector of UDP, TCP, or HTTP type. If a UDP, TCP, or HTTP connector is specified in the collector at the 'Transport' step, then at the 'Event parsing' step, you can specify multiple IP addresses on the 'Parsing settings' tab and choose the normalizer that you want to use for events coming from the specified addresses. The following types of normalizers are available: `json`, `cef`, `regexp`, `syslog`, `csv`, `kv`, `xml`. For normalizers of the `syslog` and `regexp` types, you can specify extra normalization conditions depending on the value of the DeviceProcessName field.

A normalizer is created in several steps:

1. **Preparing to create a normalizer**

   A normalizer can be created in the OSMP Console:

   - In the **Monitoring** → **Resources** → **Normalizers** section.
   - When creating a collector, at the **Event parsing** step.

   Then parsing rules must be created in the normalizer.

2. **Creating the main parsing rule for an event**

   The main parsing rule is created by clicking the **Add event parsing** button. This opens the **Event parsing** window, where you can specify the settings of the main parsing rule:

   - Specify event parsing settings.
   - Specify event enrichment settings.

   The main parsing rule for an event is displayed in the normalizer as a dark circle. You can view or modify the settings of the main parsing rule by clicking this circle. When you hover the mouse over the circle, a plus sign is displayed. Click it to add the parsing rules.

   The name of the main parsing rule is used in KUMA as the normalizer name.

3. **Creating additional event parsing rules**

   Clicking the plus icon that is displayed when you hover the mouse over the circle or the block corresponding to the normalizer opens the **Additional event parsing** window where you can specify the settings of the additional parsing rule:

   - Specify the conditions for sending data to the new normalizer.
   - Specify event parsing settings.
   - Specify event enrichment settings.

   The additional event parsing rule is displayed in the normalizer as a dark block. The block displays the triggering conditions for the additional parsing rule, the name of the additional parsing rule, and the event field. When this event field is available, the data is passed to the normalizer. Click the block of the additional parsing rule to view or modify its settings.

   If you hover the mouse over the additional normalizer, a plus button appears. You can click this button to create a new additional event parsing rule. To delete a normalizer, click the button with the trash icon.

4. **Completing the creation of the normalizer**

   To finish the creation of the normalizer, click **Save**.

In the upper right corner, in the search field, you can search for additional parsing rules by name.

For normalizer resources, you can enable the display of control characters in all input fields except the **Description** field.

> If, when changing the settings of a collector resource set, you change or delete conversions in a normalizer connected to it, the edits will not be saved, and the normalizer itself may be corrupted. If you need to modify conversions in a normalizer that is already part of a service, the changes must be made directly to the normalizer under **Monitoring** → **Resources** → **Normalizers** in OSMP Console.

Event parsing settings

You can configure the rules for converting incoming events to the KUMA format when creating event parsing rules in the normalizer settings window, on the **Normalization scheme** tab.

To define the event parsing settings:

1. In the **Name** field (required), enter the unique name of the parsing rule. Must contain 1 to 128 Unicode characters. The name of the main parsing rule is used as the name of the normalizer.

2. In the **Tenant** field (required), enter the name of the tenant that owns the resource.

   This setting is not available for extra parsing rules.

3. In the **Parsing method** drop-down list, select the type of events to receive. Depending on your choice, you can use the preconfigured rules for matching event fields or set your own rules. When you select some of the parsing methods, additional settings fields may need to be filled.

   Available parsing methods:

   - **json**

     This parsing method is used to process JSON data where each object, including its nested objects, occupies a single line in a file.

     When processing files with hierarchically structured data, you can reference the fields of nested objects using the dot notation. For example, the `username` parameter from the string `"user": {"username": "system: node: example-01"}` can be accessed by using the `user.username` query.

     Files are processed line by line. Multi-line objects with nested structures may be normalized incorrectly.

     In complex normalization schemes where additional normalizers are used, all nested objects are processed at the first normalization level, except for cases when the extra normalization conditions are not specified and, therefore, the event being processed is passed to the extra normalizer in its entirety.

     You can use `\n` and `\r\n` as newline characters. Strings must be UTF-8 encoded.

     If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

   - **cef**

     This parsing method is used to process CEF data.

If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**.

- **regexp**

  This parsing method is used to create custom rules for processing data in a format using regular expressions.

  You must add a regular expression (RE2 syntax) with named capturing groups to the field under **Normalization**. The name of the capturing group and its value are considered the field and value of the raw event that can be converted to an event field in KUMA format.

  To add event handling rules:

  1. If necessary, copy an example of the data you want to process to the **Event examples** field. We recommend completing this step.

  2. In the field under **Normalization**, add a RE2 regular expression with named capturing groups, for example, `"(?P<name>regexp)"`. The regular expression added to the field under **Normalization** must exactly match the event. When designing the regular expression, we recommend using special characters that match the starting and ending positions of the text: `^`, `$`.

     You can add multiple regular expressions or remove regular expressions. To add a regular expression, click **Add regular expression**. To remove a regular expression, click the delete icon ✕ next to it.

  3. Click the **Copy field names to the mapping table** button.

     Capture group names are displayed in the **KUMA field** column of the **Mapping** table. You can select the corresponding KUMA field in the column opposite each capturing group. If you followed the CEF format when naming the capturing groups, you can use automatic CEF mapping by selecting the **Use CEF syntax for normalization** check box.

  Event handling rules are added.

- **syslog**

  This parsing method is used to process data in syslog format.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**.

  To parse events in rfc5424 format with a `structured-data` section, in the **Keep extra fields** drop-down list, select **Yes**. This makes the values from the `structured-data` section available in the `Extra` fields.

- **csv**

  This parsing method is used to create custom rules for processing CSV data.

  When choosing this parsing method, you must specify the separator of values in the string in the **Delimiter** field. Any single-byte ASCII character can be used as a delimiter for values in a string.

- **kv**

This parsing method is used to process data in key-value pair format. Available parsing method settings are listed in the table below.

Available parsing method settings

Setting	Description
Pair delimiter	The character used to separate key-value pairs. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Value delimiter** field.
Value delimiter	The character used to separate a key from its value. You can specify any single-character (1 byte) value. The specified value must not match the value specified in the **Pair delimiter** field.

- **xml**

This parsing method is used to process XML data in which each object, including nested objects, occupies a single line in a file. Files are processed line by line.

If you want to send the raw event for advanced normalization, at each nesting level in the **Advanced event parsing** window, select **Yes** in the **Keep raw event** drop-down list.

If you select this parsing method, under **XML attributes**, you can specify the key XML attributes to be extracted from tags. If an XML structure has multiple XML attributes with different values in the same tag, you can identify the necessary value by specifying the key of the value in the **Source** column of the **Mapping** table.

To add key XML attributes:

1. Click **+ Add field**.

2. This opens a window; in that window, specify the path to the XML attribute.

You can add multiple XML attributes or remove XML attributes. To remove an individual XML attribute, click the delete icon ✕ next to it. To remove all XML attributes, click **Reset**.

If XML key attributes are not specified, then in the course of field mapping the unique path to the XML value will be represented by a sequence of tags.

Tag numbering

Starting with KUMA 2.1.3, you can use automatic tag numbering in XML events. This lets you parse an event with the identical tags or unnamed tags, such as `<Data>`.

As an example, we will number the tags of the `EventData` attribute of the **Microsoft Windows PowerShell event ID 800**.

```
<Event xmlns="http://schemas .microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-ActiveDirectory_DomainService" Guid="
    {0e8478c5-3605-4e8c-8497-1e730c959516}" EventSourceName="NTDS" />
    <EventID Qualifiers="0000">0000</EventID>
    <Version>@</Version>
    <Level>4</Level>
    <Task>15</Task>
```

```
    <Opcode>0</Opcode>
    <Keywords >0x8080000000000000</Keywords>
    <TimeCreated SystemTime="2000-01-01T00:00:00.659495900Z" />
    <EventRecordID>55647</EventRecordID>
    <Correlation />
    <Execution ProcessID="1" ThreadID="1" />
    <Channel>service</Channel>
    <Computer>computer</Computer>
    <Security UserID="0000" />
  </System>
  <EventData>
    <Data>583</Data>
    <Data>36</Data>
    <Data>192.168.0.1:5084</Data>
    <Data>level</Data>
    <Data>name, lDAPDisplayName</Data>
    <Data />
    <Data>5545</Data>
    <Data>3</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>0</Data>
    <Data>15</Data>
    <Data>none</Data>
  </EventData>
</Event>
```

To parse events with identical tags or unnamed tags, you need to configure tag numbering and data mapping for numbered tags with KUMA event fields.

KUMA 3.0.x supports using XML attributes and tag numbering at the same time in the same extra normalizer. If an XML attribute contains unnamed tags or identical tags, we recommend using tag numbering. If the XML attribute contains only named tags, we recommend using XML attributes.

To use XML attributes and tag numbering in extra normalizers, you must sequentially enable the **Keep raw event** setting in each extra normalizer along the path that the event follows to the target extra normalizer, and in the target extra normalizer itself.

For an example of how tag numbering works, you can refer to the **MicrosoftProducts** normalizer. The **Keep raw event** setting is enabled sequentially in both **AD FS** and **424** extra normalizers.

To set up the parsing of events with unnamed or identical tags:

1. Open an existing normalizer or create a new normalizer.

2. In the **Basic event parsing** window of the normalizer, in the **Parsing method** drop-down list, select **xml**.

3. In the **Tag numbering** field, click **+ Add field**.

4. In the displayed field, enter the full path to the tag to whose elements you want to assign a number, for example, `Event.EventData.Data`. The first tag gets number 0. If the tag is empty, for example, `<Data />`, it is also assigned a number.

5. To configure data mapping, under **Mapping**, click **+ Add row** and do the following:

   a. In the displayed row, in the **Source** field, enter the full path to the tag and the index of the tag. For example, for the Microsoft Windows PowerShell event ID 800 from the example above, the full paths to tags and tag indices are as follows:

      - `Event.EventData.Data.0,`
      - `Event.EventData.Data.1,`
      - `Event.EventData.Data.2` and so on.

   b. In the **KUMA field** drop-down list, select the field in the KUMA event that will receive the value from the numbered tag after parsing.

6. Save changes in one of the following ways:

   - If you created a new normalizer, click **Save**.
   - If you edited an existing normalizer, in the collector to which the normalizer is linked, click **Update configuration**.

Parsing is configured.

- **netflow5**

This parsing method is used to process data in the NetFlow v5 format.

If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **netflow5** parsing method is selected for the main parsing, extra normalization is not available.

> The default mapping rules for the **netflow5** parsing method do not specify the protocol type in KUMA event fields. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- **netflow9**

This parsing method is used to process data in the NetFlow v9 format.

If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **netflow9** parsing method is selected for the main parsing, extra normalization is not available.

> The default mapping rules for the **netflow9** parsing method do not specify the protocol type in KUMA event fields. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- **sflow5**

  This parsing method is used to process data in sflow5 format.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **sflow5** parsing method is selected for the main parsing, extra normalization is not available.

- **ipfix**

  This parsing method is used to process IPFIX data.

  If you select this parsing method, you can use the predefined rules for converting events to the KUMA format by clicking **Apply default mapping**. If the **ipfix** parsing method is selected for the main parsing, extra normalization is not available.

  The default mapping rules for the **ipfix** parsing method do not specify the protocol type in KUMA event fields. When parsing data in NetFlow format, on the **Enrichment** normalizer tab, you must create a **constant** data enrichment rule that adds the `netflow` value to the `DeviceProduct` target field.

- **sql**—this method becomes available only when using a sql type connector.

  The normalizer uses this parsing method to process data obtained by making a selection from the database.

4. In the **Keep raw event** drop-down list, specify whether to store the original raw event in the newly created normalized event. Available values:

   - **Don't save**—do not save the raw event. This is the default setting.

   - **Only errors**—save the raw event in the `Raw` field of the normalized event if errors occurred when parsing it. This value is convenient to use when debugging a service. In this case, every time an event has a non-empty `Raw` field, you know there was a problem.

     If fields containing the names `*Address` or `*Date*` do not comply with normalization rules, these fields are ignored. No normalization error occurs in this case, and the values of the fields are not displayed in the `Raw` field of the normalized event even if the **Keep raw event** → **Only errors** option was selected.

   - **Always**—always save the raw event in the `Raw` field of the normalized event.

   This setting is not available for extra parsing rules.

5. In the **Keep extra fields** drop-down list, choose whether you want to store the raw event fields in the normalized event if no mapping rules have been configured for them (see below). The data is stored in the `Extra` event field. Normalized events can be searched and filtered based on the data stored in the `Extra` field.

   Filtering based on data from the Extra event field

Conditions for filters based on data from the `Extra` event field:

- Condition—**If**.
- Left operand—**event field**.
- In this event field, you can specify one of the following values:
  - `Extra` field.
  - Value from the `Extra` field in the following format:

    `Extra.<field name>`

    For example, `Extra.app`.

    You must specify the value manually.
  - Value from the array written to the `Extra` field in the following format:

    `Extra.<field name>.<array element>`

    For example, `Extra.array.0`.

    The values in the array are numbered starting from 0. You must specify the value manually. To work with a value in the `Extra` field at a depth of 3 and lower, you must use backticks ``, for example, `` `Extra.lev1.lev2.lev3` ``.
- Operator – =.
- Right operand—**constant**.
- Value—the value by which you need to filter events.

By default, fields are not saved.

6. In the **Description** field, specify the resource description: up to 4,000 Unicode characters.

   This setting is not available for extra parsing rules.

7. In the **Event examples** field, you can provide an example of data that you want to process.

   This setting is not available for the following parsing methods: **netflow5**, **netflow9**, **sflow5**, **ipfix**, **sql**.

   The **Event examples** field is populated with data obtained from the raw event if the event was successfully parsed and the type of data obtained from the raw event matches the type of the KUMA field.

   For example, the value "192.168.0.1" enclosed in quotation marks is not displayed in the SourceAddress field, in this case the value 192.168.0.1 is displayed in the **Event examples** field.

8. In the **Mapping** table, configure the mapping of raw event fields to fields of the event in KUMA format:

   a. In the **Source** column, provide the name of the raw event field that you want to convert into the KUMA event field.

      For details about the field format, refer to the Normalized event data model article. For a description of the mapping, refer to the Mapping fields of predefined normalizers article.

      Clicking the 🔧 button next to the field names in the **Source** column opens the **Conversion** window, in which you can click the **Add conversion** button to create rules for modifying the original data before they are written to the KUMA event fields.

      Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:

  - **decodeHexString**—used to convert a HEX string to text.

  - **decodeBase64String**—used to convert a Base64 string to text.

  - **decodeBase64URLString**—used to convert a Base64url string to text.

When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

In the **Conversion** window, you can swap the added rules by dragging them by the ⠿ icon; you can also delete them using the × icon.

b. In the **KUMA field** column, select the required KUMA event field from the drop-down list. You can search for fields by entering their names in the field.

Recommendations concerning the KUMA field column

We recommend that you configure the mapping for the following KUMA fields. Otherwise, you will not be able to view observables in alert details and incident details.

The recommended KUMA fields depend on the observable types:

- For observables of the MD5 hash and SHA256 types:
  - FileHash
- For observables of the URL type:
  - RequestUrl
- For observables of the IP address type:
  - DeviceCustomIPv6Address1
  - DeviceCustomIPv6Address2
  - DeviceCustomIPv6Address3
  - DeviceCustomIPv6Address4

- DestinationTranslatedAddress
- DeviceTranslatedAddress
- DestinationAddress
- DeviceAddress
- SourceTranslatedAddress
- SourceAddress

- For observables of the Domain name type:
  - DestinationDnsDomain
  - DeviceDnsDomain
  - DeviceNtDomain
  - DestinationNtDomain
  - SourceDnsDomain
  - SourceNtDomain
- For observables of the UserName type:
  - DestinationUserName
  - SourceUserName
- For observables of the HostName type:
  - DestinationHostName
  - DeviceHostName
  - SourceHostName

c. If the name of the KUMA event field selected at the previous step begins with `DeviceCustom*` or `Flex*`, you can add a unique custom label in the **Label** field.

New table rows can be added by clicking the **Add row** button. Rows can be deleted individually by clicking the ✕ button or all at once by clicking the **Clear all** button.

If you want KUMA to enrich events with asset information, and the asset information to be available in the alert card when a correlation rule is triggered, in the **Mapping** table, configure a mapping of host address and host name fields depending on the purpose of the asset. For example, the mapping can apply to SourceAddress and SourceHostName, or DestinationAddress and DestinationHostName fields. As a result of enrichment, the event card includes a SourceAssetID or DestinationAssetID field, and a link to the asset card. Also, as a result of enrichment, asset information is available in the alert card.

If you have loaded data into the **Event examples** field, the table will have an **Examples** column containing examples of values carried over from the raw event field to the KUMA event field.

If the size of the KUMA event field is less than the length of the value placed in it, the value is truncated to the size of the event field.

Extended event schema

When normalizing events, extended event schema fields can be used in addition to standard KUMA event schema fields. Information about the types of extended event schema fields is shown in the table below.

Using many unique fields of the extended event schema can reduce the performance of the system, increase the amount of disk space required for storing events, and make the information difficult to understand.

We recommend consciously choosing a minimal set of additional fields of the extended event schema that you want to use in normalizers and correlation.

To use extended event schema fields:

- Open an existing event normalizer or create a new event normalizer.
- Specify the basic settings of the normalizer.
- Click **Add row**.
- For the **Source** setting, enter the name of the source field in the raw event.
- For the **KUMA field** setting, enter the name of the extended event schema field that you are creating (see the table below). You can also use an existing field of the extended event schema.

  Fields of the extended data model of normalized events:

Field name Specified in the KUMA field setting	Data type	Availability in the normalizer	Description
S.<`field name`>	String	All types	Field of the `String` type
N.<`field name`>	Number	All types	Field of the `Number` type
F.<`field name`>	Float	All types	Field of the `Float` type
SA.<`field name`>	Array of strings	KV, JSON	Field of the `Array of strings` type. The order of the array elements is the same as the order of the elements of the raw event.
NA.<`field name`>	Array of integers	KV, JSON	A field of the `Array of integers` type. The order of the array elements is the same as the order of the elements of the raw event.
FA.<`field name`>	Array of floats	KV, JSON	Field of the `Array of floats` type. The order of the array elements is the same as the order of the elements of the raw event.

The prefixes `S.`, `N.`, `F.`, `SA.`, `NA.`, `FA.` are required when creating fields of the extended event schema; the prefixes must be strictly uppercase.

Replace < `field name` > with the field name. You may use letters of the English alphabet and numerals in the field name. The space character is not allowed.

- Click **OK**.
- Click **Save** to finish editing the event normalizer.

The normalizer is saved, and the additional field is created. After saving the normalizer, the additional field can be used in other normalizers.

Note: If the data in the fields of the raw event does not match the type of the KUMA field, the value is not saved during the normalization of events. For example, the string `test` cannot be written to the DeviceCustomNumber1 KUMA field of the `Number` type.

If you want to minimize the load on the storage server when searching events, preparing reports, and performing other operations on events in storage, use KUMA event schema fields as your first preference, extended event schema fields as your second preference, and the Extra fields as your last resort.

Extended event schema

You can use the extended event schema fields in normalizers for normalizing events and in other KUMA resources, for example, as widget fields or to filter and search for events. You can view the list of all extended event schema fields that exist in KUMA in the **Settings** → **KUMA** → **Open KUMA settings** → **Extended fields set** section. The list of extended event schema fields is the same for all tenants.

Only users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Read shared resources, and Manage shared resources roles can view the table of extended event schema fields.

The **Extended event schema fields** table contains the following information:

- **Type**—Data type of the extended event schema field.

  Possible data types:

Type	Data type
S	String.
N	Number.
F	Floating point number.
SA	Array of strings. The order of the array elements is the same as the order of the elements of the raw event.
NA	Array of integers. The order of the array elements is the same as the order of the elements of the raw event.
FA	Array of floats. The order of the array elements is the same as the order of the elements of the raw event.

- **Field name**—Name of the extended event schema field, without a type.

  You can click the name to edit the settings of the extended event schema field.

- **Status**—Whether the extended event schema field can be used in resources.

  You can **Enable** or **Disable** the toggle switch to allow or forbid using this extended event schema field in new resources. However, a disabled field is still used in resource configurations that are already operational, until you manually remove the field from the configuration; the field also remains available in the list of table columns in the **Events** section for managing old events.

  Only a user with the General administrator role can disable an extended event schema field.

- **Update date**—Date and time of the last modification of the extended event schema field.

- **Created by**—Name of the user that created the extended event schema field.

- **Dependencies**—Number of KUMA resources, dashboard layouts, reports, presets, and field sets for searching event sources that use the extended event schema field.

  You can click the number to open a pane with a table of all resources and other KUMA entities that are using this field. For each dependency, the table displays the name, tenant (only for resources), and type. Dependencies in the table are sorted by name. Clicking the name of a dependency takes you to its page (except for dashboard layouts, presets, and saved user queries).

  You can view the dependencies of an extended event schema field only for resources and entities to whose tenants you have access. If you do not have access to the tenant, its resources are not displayed in the table, but still count towards the number of dependencies.

- **Description**—Text description of the field.

By default, the table of extended event schema fields is sorted by update date in descending order. If necessary, you can sort the table by clicking a column heading and selecting **Ascending** or **Descending**; you can also use context search by field name.

By default, the following service extended event schema fields are automatically added to KUMA:

- `KL_EventRoute`, type `S` for storing information about the route of the event.

  You can use this field in normalizers, as a key or value in active lists, in enrichment rules, as a query field in data collection and analysis rules, in correlation rules. You cannot use this field to detect event sources.

- The following fields are added to a correlation event:

  - `KL_CorrelationRulePriority`, type `N`
  - `KL_SourceAssetDisplayName`, type `S`
  - `KL_DestinationAssetDisplayName`, type `S`
  - `KL_DeviceAssetDisplayName`, type `S`
  - `KL_SourceAccountDisplayName`, type `S`
  - `KL_DestinationAccountDisplayName`, type `S`

  You cannot use this service fields to search for events.

You cannot edit, delete, export, or disable service fields. All extended event schema fields with the `KL_` prefix are service fields and can be managed only from Kaspersky servers. We do not recommend using the `KL_` prefix when adding new extended event schema fields.

Adding extended event schema fields

Users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Manage shared resources roles can add new extended event schema fields.

To add an extended event schema field:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Other**, click **Extended fields set**.

   This opens the **Extended fields set** window.

3. Click the **Add** button in the upper part of the table.

   This opens the **Create extended schema** pane.

4. Enable or disable the **Status** toggle switch to enable or disable this extended event schema field for resources.

   The toggle switch is turned on by default. A disabled field remains available in the list of table columns in the **Events** section for managing old events.

5. In the **Type** field, select the data type of the extended event schema field.

   Possible data types

Type	Data type
S	String.
N	Number.
F	Floating point number.
SA	Array of strings. The order of the array elements is the same as the order of the elements of the raw event.
NA	Array of integers. The order of the array elements is the same as the order of the elements of the raw event.
FA	Array of floats. The order of the array elements is the same as the order of the elements of the raw event.

6. In the **Name** field, specify the name of the extended event schema field.

   Consider the following when naming extended event schema fields:

   - The name must be unique within the KUMA instance.
   - Names are case-sensitive. For example, `Field_name` and `field_name` are different names.
   - You can use Latin, Cyrillic characters and numerals. Spaces or " ~ ` @ # $ % ^ & * ( ) + - [ ] { } | \ | / . " < > ; ! , : ? = characters are not allowed.
   - If you want to use the extended event schema fields to search for event sources, you can only use Latin characters and numerals.
   - The maximum length is 128 characters.

7. If necessary, in the **Description** field, enter a description for the extended event schema field.

   We recommend describing the purpose of the extended event schema field. Only Unicode characters are allowed in the description. The maximum length is 256 characters.

8. Click the **Save** button.

A new extended event schema field is added and displayed at the top of the table. An audit event is generated for the creation of the extended event schema field. If you have enabled the field, you can use it in normalizers and when configuring resources.

Editing extended event schema fields

Users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, Manage shared resources roles can edit existing extended event schema fields.

To edit an extended event schema field:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Other**, click **Extended fields set**.

   This opens the **Extended fields set** window.

3. Click the name of the field that you want to edit.

   This opens the **Edit extended schema** pane. This pane displays the settings of the selected field, as well as the **Dependencies** table with a list of resources, dashboard layouts, reports, presets, and sets of fields for finding event sources that use this field. Only resources to whose tenants you have access are displayed. If the field is used by resources to whose tenant you do not have access, such resources are not displayed in the table. Resources in the table are sorted by name.

   Clicking the name of a resource or entity takes you to its page (except for dashboard resources, presets, and saved user queries).

4. Make the changes you need in the available settings.

   You can edit the **Type** and **Field name** settings only if the extended event schema field does not have dependencies. You can edit the **Status** and **Description** settings for any extended event scheme field. However, a field with the **Disabled** status is still used in resource configurations that are already operational, until you manually remove the field from the configuration; the field also remains available in the list of table columns in the **Events** section for managing old events.

   Disabling an extended event schema field using the **Status** field requires the General administrator role.

5. Click the **Save** button.

The extended event schema field is updated. An audit event is generated about the modification of the field.

Importing and exporting extended event schema fields

You can add multiple new extended event schema fields at once by importing them from a JSON file. You can also export all extended event schema fields with information about them to a file, for example, to propagate the list of fields to other KUMA instances to maintain resources.

Users with the General administrator, Tenant administrator, Tier 2 analyst, Tier 1 analyst, Junior analyst, and Manage shared resources roles can import an export extended event schema fields. Users with the Read shared resources role can only export extended event schema fields.

To import extended event schema fields into KUMA from a file:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Other**, click **Extended fields set**.

   This opens the **Extended fields set** window.

3. Click the **Import** button.

4. This opens a window; in that window, select a JSON file with a list of extended event schema field objects.

   Example JSON file:

```
[
{"kind": "SA",
"name": "<fieldName1>",
"description": "<description1>",
"disabled": false},
{"kind": "N",
"name": "<fieldName2>",
"description": "<description2>",
"disabled": false},
....
{"kind": "FA",
"name": "<fieldNameX>",
"description": "<descriptionX>",
"disabled": false}
]
```

> When importing fields from a file, their names are checked for possible conflicts with fields of the same type. If a field with the same name and type already exists in KUMA, such fields are not imported from the file.

Extended event schema fields are imported from the file to KUMA. An audit event about the import of fields is generated, and a separate audit event is generated for each added field.

To export extended event schema fields to a file:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Other**, click **Extended fields set**.

   This opens the **Extended fields set** window.

3. If you want to export specific extended event schema fields:

   a. Select the check boxes in the first column of the table for the required fields.

      You cannot select service fields.

   b. Click the **Export selected** button in the upper part of the table.

4. If you want to export all extended event schema fields, click the **Export all** button in the upper part of the table.

A JSON file with a list of extended event schema field objects and information about them is downloaded.

Deleting extended event schema fields

Only a user with the General administrator role can delete extended event schema fields.

You can delete only those extended event schema fields that are not service fields, that have the **Disabled** status, and that are not used in KUMA resources and other entities (do not have dependencies). We recommend deleting extended event schema fields after enough time has passed to make sure that all events in which the field was used have been deleted from KUMA. When you delete a field, it is no longer displayed in event tips.

To delete extended event schema fields:

1. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

2. Under **Other**, click **Extended fields set**.

   This opens the **Extended fields set** window.

3. Select the check boxes in the first column of the table next to one or more fields that you want to delete.

   To select all fields, you can select the check box in the heading of the first column.

4. Click the **Delete** button in the upper part of the table.

   The **Delete** button is active only if all selected fields are disabled and have no dependencies. If at least one field is enabled or has a dependency, the button is inactive.

   If you want to delete a field that is used in at least one KUMA resource (has a dependency), but you do not have access to its tenant, the **Delete** button is active when this field is selected, but an error is displayed when you try to delete it.

The selected fields are deleted. An audit event is generated about the deletion of the fields.

Using extended event schema fields in normalizers

When using extended event schema fields, the general limit for the maximum size of an event that can be processed by the collector is the same, 4 MB. Information about the types of extended event schema fields is shown in the table below (step 6 of the instructions).

> Using many unique fields of the extended event schema can reduce the performance of the system, increase the amount of disk space required for storing events, and make the information difficult to understand.

We recommend consciously choosing a minimal set of additional fields of the extended event schema that you want to use in normalizers and correlation.

To use the fields of the extended event schema:

1. Open an existing normalizer or create a new normalizer.

2. Specify the basic settings of the normalizer.

3. Click **Add row**.

4. For the **Source** setting, enter the name of the source field in the raw event.

5. For the **KUMA field** setting, start typing the name of the extended event schema field and select the field from the drop-down list.

   The extended event schema fields in the drop-down list have names in the `<type>.<field name>` format.

6. Click the **Save** button to save the event normalizer.

The normalizer is saved with the selected extended event schema field.

> If the data in the fields of the raw event does not match the type of the KUMA field, the value is not saved during the normalization of events if type conversion cannot be performed. For example, the string `test` cannot be written to the `DeviceCustomNumber1` KUMA field of the Number type.

If you want to minimize the load on the storage server when searching events, preparing reports, and performing other operations on events in storage, use KUMA event schema fields as your first preference, extended event schema fields as your second preference, and the `Extra` fields as your last resort.

Enrichment in the normalizer

When creating event parsing rules in the normalizer settings window, on the **Enrichment** tab, you can configure the rules for adding extra data to the fields of the normalized event using enrichment rules. These enrichment rules are stored in the settings of the normalizer where they were created.

Enrichments are created by clicking the **Add enrichment** button. There can be more than one enrichment rule. You can delete enrichment rules by clicking the × button.

Settings available in the enrichment rule settings block:

- **Source kind** (required)—drop-down list for selecting the type of enrichment. Depending on the selected type, you may see advanced settings that will also need to be completed.

  Available Enrichment rule source types:

  - **constant**

    This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

    Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

    If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

    If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- **dictionary**

  This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

  If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

  Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

  If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the " | " character when the dictionary is queried.

  Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **table**

  This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

  When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, click the **Add field** button to select the event fields whose values are used for dictionary entry selection.

  In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

  - In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
  - In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

  New table rows can be added by clicking the **Add new element** button. Columns can be deleted by clicking the ✕ button.

- **event**

  This type of enrichment is used when you need to write a value from another event field to the current event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Target field	The KUMA event field that you want to populate with the data.
Source field	The event field whose value is written to the target field.

Clicking 🔧 opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the KUMA event fields. You can reorder and delete created rules. To change the position of a rule, click ⠿ next to it. To delete a rule, click ✕ next to it.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.
- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
  - **Replace chars** specifies the sequence of characters to be replaced.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.
- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:
  - **Expression** is the RE2 regular expression whose results you want to replace.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

When using enrichment of events that have **event** selected as the **Source kind** and the extended event schema fields are used as arguments, the following special considerations apply:

- If the source extended event schema field has the `Array of strings` type, and the target extended event schema field has the `String` type, the values are written to the target extended event schema field in TSV format.

  Example: The `SA.StringArray` extended event schema field contains values: `"string1"`, `"string2"`, `"string3"`. An event enrichment operation is performed. The result of the event enrichment operation is written to the `DeviceCustomString1` extended event schema field. The `DeviceCustomString1` extended event schema field contains values: `["string1", "string2", "string3"]`.

- If the source and target extended event schema fields have the `Array of strings` type, values of the source extended event schema field are added to the values of the target extended event schema field, and the `","` character is used as the delimiter character.

  Example: The `SA.StringArrayOne` field of the extended event scheme contains the `["string1", "string2", "string3"]` values, and the `SA.StringArrayTwo` field of the extended event scheme contains the `["string4", "string5", "string6"]` values. An event enrichment operation is performed. The result of the event enrichment operation is written to the `SA.StringArrayTwo` field of the extended event scheme. The `SA.StringArrayTwo` extended event schema field contains values: `["string4", "string5", "string6", "string1", "string2", "string3"]`.

- **template**

  This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

  If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

  - `{{.SA.StringArrayOne}}`
  - `{{- range $index, $element := . SA.StringArrayOne -}}`
    `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

  To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

  ```
  template {{toString .SA.StringArray}}
  ```

- **Target field** (required)—drop-down list for selecting the KUMA event field that should receive the data.

  This setting is not available for the enrichment source of the **Table** type.

Conditions for forwarding data to an extra normalizer

When creating additional event parsing rules, you can specify the conditions. When these conditions are met, the events are sent to the created parsing rule for processing. Conditions can be specified in the **Additional event parsing** window, on the **Extra normalization conditions** tab. This tab is not available for the basic parsing rules.

Available settings:

- **Use raw event** — If you want to send a raw event for extra normalization, select **Yes** in the **Keep raw event** drop-down list. The default value is **No**. We recommend passing a raw event to normalizers of json and xml types. If you want to send a raw event for extra normalization to the second, third, etc nesting levels, at each nesting level, select **Yes** in the **Keep raw event** drop-down list.

- **Field to pass into normalizer**—indicates the event field if you want only events with fields configured in normalizer settings to be sent for additional parsing.

  If this field is blank, the full event is sent to the extra normalizer for processing.

- Set of filters—used to define complex conditions that must be met by the events received by the normalizer.

  You can click the **Add condition** button to add a string containing fields for identifying the condition (see below).

  You can click the **Add group** button to add a group of filters. Group operators can be switched between **AND**, **OR**, and **NOT**. You can add other condition groups and individual conditions to filter groups.

  You can swap conditions and condition groups by dragging them by the ⠿ icon; you can also delete them using the ✕ icon.

Filter condition settings:

- **Left operand** and **Right operand**—used to specify the values to be processed by the operator.

  In the left operand, you must specify the source field of events coming into the normalizer. For example, if the eventType - DeviceEventClass mapping is configured in the **Basic event parsing** window, then in the **Additional event parsing** window on the **Extra normalization conditions** tab, you must specify eventType in the left operand field of the filter. Data is processed only as text strings.

- Operators:

  - **=** – full match of the left and right operands.

  - **startsWith** – the left operand starts with the characters specified in the right operand.

  - **endsWith** – the left operand ends with the characters specified in the right operand.

  - **match** – the left operand matches the regular expression (RE2) specified in the right operand.

  - **in** – the left operand matches one of the values specified in the right operand.

The incoming data can be converted by clicking the 🔧 button. The **Conversion** window opens, where you can click the **Add conversion** button to create the rules for converting the source data before any actions are performed on them. In the **Conversion** window, you can swap the added rules by dragging them by the ⠿ icon; you can also delete them using the ✕ icon.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.

- **lower**—is used to make all characters of the value lowercase

- **upper**—is used to make all characters of the value uppercase

- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.

- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:

  - **Replace chars** specifies the sequence of characters to be replaced.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.

- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.

- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:

  - **Expression** is the RE2 regular expression whose results you want to replace.

  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:

  - **decodeHexString**—used to convert a HEX string to text.

  - **decodeBase64String**—used to convert a Base64 string to text.

  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

Supported event sources

KUMA supports the normalization of events coming from systems listed in the table below. Normalizers for these systems are included in the distribution kit.

Supported event sources

System name	Normalizer name	Type	Normalizer description
1C EventJournal	[OOTB] 1C EventJournal Normalizer	xml	Designed for processing the event log of the 1C system. The event source is the 1C log.
1C TechJournal	[OOTB] 1C TechJournal Normalizer	regexp	Designed for processing the technology event log. The event source is the 1C technology log.
Absolute Data and Device Security (DDS)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
AhnLab Malware Defense System (MDS)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Ahnlab UTM	[OOTB] Ahnlab UTM	regexp	Designed for processing events from the Ahnlab system. The event sources is system logs, operation logs, connections, the IPS module.
AhnLabs MDS	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Alcatel AOS-W	[OOTB] Alcatel AOS-W syslog	regexp	Designed for processing some of the events received from Alcatel AOS-W 6.4 via Syslog.
Alcatel Network Switch	[OOTB] Alcatel Network Switch syslog	Syslog	Designed for processing certain types of events received from Alcatel network switches via Syslog.
Apache Cassandra	[OOTB] Apache Cassandra file	regexp	Designed for processing events from the logs of the Apache Cassandra database version 4.0.
Aruba ClearPass	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.

System name	Normalizer name	Type	Normalizer description
Atlassian Confluence	[OOTB] Atlassian Jira Confluence file	regexp	Designed for processing events of Atlassian Jira, Atlassian Confluence systems (Jira 9.12, Confluence 8.5) stored in files.
Atlassian Jira	[OOTB] Atlassian Jira Confluence file	regexp	Designed for processing events of Atlassian Jira, Atlassian Confluence systems (Jira 9.12, Confluence 8.5) stored in files.
Avanpost FAM	[OOTB] Avanpost FAM syslog	regexp	Designed for processing events of the Avanpost Federated Access Manager (FAM) 1.9 received via Syslog.
Avanpost IDM	[OOTB] Avanpost IDM syslog	regexp	Designed for processing events of the Avanpost IDM system received via Syslog.
Avanpost PKI	[OOTB] Avanpost PKI syslog CEF	Syslog	Designed for processing events received from Avanpost PKI 6.0 in CEF format via Syslog.
Avaya Aura Communication Manager	[OOTB] Avaya Aura Communication Manager syslog	regexp	Designed for processing some of the events received from Avaya Aura Communication Manager 7.1 via syslog.
Avigilon Access Control Manager (ACM)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Ayehu eyeShare	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Arbor Pravail	[OOTB] Arbor Pravail syslog	Syslog	Designed for processing events of the Arbor Pravail system received via syslog.
Aruba Aruba AOS-S	[OOTB] Aruba Aruba AOS-S syslog	regexp	Designed for processing certain types of events received from Aruba network devices with Aruba AOS-S 16.10 firmware via syslog. The normalizer supports the following types of events: accounting events, ACL events, ARP protect events, authentication events, console events, loop protect events.
Barracuda Cloud Email Security Gateway	[OOTB] Barracuda Cloud Email Security Gateway syslog	regexp	Designed for processing events from Barracuda Cloud Email Security Gateway via syslog.
Barracuda Networks NG Firewall	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Barracuda Web Security Gateway	[OOTB] Barracuda Web Security Gateway syslog	Syslog	Designed for processing some of the events received from Barracuda Web Security Gateway 15.0 via Syslog.
BeyondTrust Privilege Management Console	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BeyondTrust's BeyondInsight	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Bifit Mitigator	[OOTB] Bifit Mitigator Syslog	Syslog	Designed for processing events from the DDOS Mitigator protection system received via Syslog.
Bloombase StoreSafe	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BMC CorreLog	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Bricata ProAccel	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Brinqa Risk Analytics	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Broadcom Symantec Advanced Threat Protection (ATP)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.

System name	Normalizer name	Type	Normalizer description
Broadcom Symantec Endpoint Protection	[OOTB] Broadcom Symantec Endpoint Protection	regexp	Designed for processing events from the Symantec Endpoint Protection system.
Broadcom Symantec Endpoint Protection Mobile	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Broadcom Symantec Threat Hunting Center	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Brocade Fabric OS	[OOTB] Brocade Fabric OS syslog	Syslog	Designed for processing events of Brocade Fabric 9.1 received via syslog.
Canonical LXD	[OOTB] Canonical LXD syslog	Syslog	Designed for processing events received via Syslog from the Canonical LXD system version 5.18.
Checkpoint	[OOTB] Checkpoint syslog	Syslog	[OOTB] Checkpoint syslog — designed for processing events received from the Checkpoint R81 firewall via the Syslog protocol.
Cisco Access Control Server (ACS)	[OOTB] Cisco ACS syslog	regexp	Designed for processing events of the Cisco Access Control Server (ACS) system received via Syslog.
Cisco ASA	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for certain events of Cisco ASA and Cisco IOS devices received via Syslog.
Cisco Email Security Appliance (WSA)	[OOTB] Cisco WSA AccessFile	regexp	Designed for processing the event log of the Cisco Email Security Appliance (WSA) proxy server, the access.log file.
Cisco ESA syslog	[OOTB] Cisco ESA syslog	Syslog	Designed for processing certain types of events received from Alcatel network switches via Syslog.
Cisco Firepower Threat Defense	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for processing events for network devices: Cisco ASA, Cisco IOS, Cisco Firepower Threat Defense (version 7.2) received via Syslog.
Cisco Identity Services Engine (ISE)	[OOTB] Cisco ISE syslog	regexp	Designed for processing events of the Cisco Identity Services Engine (ISE) system received via Syslog.
Cisco IOS	[OOTB] Cisco ASA and IOS syslog	Syslog	Designed for certain events of Cisco ASA and Cisco IOS devices received via Syslog.
Cisco Netflow v5	[OOTB] NetFlow v5	netflow5	Designed for processing events from Cisco Netflow version 5.
Cisco NetFlow v9	[OOTB] NetFlow v9	netflow9	Designed for processing events from Cisco Netflow version 9.
Cisco Prime	[OOTB] Cisco Prime syslog	Syslog	Designed for processing events of the Cisco Prime system version 3.10 received via Syslog.
Cisco Secure Email Gateway (SEG)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Cisco Secure Firewall Management Center	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Cisco WLC	[OOTB] Cisco WLC syslog	regexp	Normalizer for some types of events received from Cisco WLC network devices (2500 Series Wireless Controllers, 5500 Series Wireless Controllers, 8500 Series Wireless Controllers, Flex 7500 Series Wireless Controllers) via Syslog.

System name	Normalizer name	Type	Normalizer description
Cisco WSA	[OOTB] Cisco WSA file, [OOTB] Cisco WSA syslog	regexp	[OOTB] Cisco WSA file. This normalizer is designed for processing the event log of the Cisco WSA proxy server (versions 14.2, 15.0). The normalizer supports processing events generated using the following template: %t %e %a %w/%h %s %2r %A %H/%d %c %D %Xr %?BLOCK_SUSPECT_USER_AGENT,MONITOR_SUSPECT_USER_AGENT?%<User-Agent:%!%-%. %) %q %k %u %m [OOTB] Cisco WSA syslog. This normalizer is designed for processing events received from the Cisco WSA system (version 15.0) via Syslog.
Citrix NetScaler	[OOTB] Citrix NetScaler syslog	regexp	Designed for processing events received from the Citrix NetScaler 13.7 load balancer, Citrix ADC NS13.0.
Claroty Continuous Threat Detection	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CloudPassage Halo	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Codemaster Mirada	[OOTB] Codemaster Mirada syslog	Syslog	Designed for processing events of the Codemaster Mirada system received via Syslog.
CollabNet Subversion Edge	[OOTB] CollabNet Subversion Edge syslog	Syslog	Designed for processing events received from the Subversion Edge (version 6.0.2) system via Syslog.
CommuniGate Pro	[OOTB] CommuniGate Pro	regexp	Designed to process events of the CommuniGate Pro 6.1 system sent by the KUMA agent via TCP.
Corvil Network Analytics	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Cribl Stream	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CrowdStrike Falcon Host	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CyberArk Privileged Threat Analytics (PTA)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
CyberPeak Spektr	[OOTB] CyberPeak Spektr syslog	Syslog	Designed for processing events of the CyberPeak Spektr system version 3 received via Syslog.
Cyberprotect Cyber Backup	[OOTB] Cyberprotect Cyber Backup SQL	sql	Designed for processing events received by the connector from the database of the Cyber Backup system (version 16.5).
DeepInstinct	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Delinea Secret Server	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Dell Network Switch	[OOTB] Dell Network Switch syslog	regexp	Designed for processing certain types of events received from Dell network switches via Syslog.
Digital Guardian Endpoint Threat Detection	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
BIND DNS server	[OOTB] BIND Syslog [OOTB] BIND file	Syslog regexp	[OOTB] BIND Syslog is designed for processing events of the BIND DNS server received via Syslog. [OOTB] BIND file is designed for processing event logs of the BIND DNS server.
Docsvision	[OOTB] Docsvision syslog	Syslog	Designed for processing audit events received from the Docsvision system via Syslog.
Dovecot	[OOTB] Dovecot Syslog	Syslog	Designed for processing events of the Dovecot mail server received via Syslog. The event source is POP3/IMAP logs.

System name	Normalizer name	Type	Normalizer description
Dragos Platform	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Dr.Web Enterprise Security Suite	[OOTB] Syslog-CEF	Syslog	Designed for processing Dr.Web Enterprise Security Suite 13.0.1 events in the CEF format.
EclecticIQ Intelligence Center	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Edge Technologies AppBoard and enPortal	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Eltex ESR	[OOTB] Eltex ESR syslog	Syslog	Designed to process part of the events received from Eltex ESR network devices via Syslog.
Eltex MES	[OOTB] Eltex MES syslog	regexp	Designed for processing events received from Eltex MES network devices via Syslog (supported device models: MES14xx, MES24xx, MES3708P).
ESET Protect	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Extreme Networks Summit Wireless Controller	[OOTB] Extreme Networks Summit Wireless Controller	regexp	Normalizer for certain audit events of the Extreme Networks Summit Wireless Controller (model: WM3700, firmware version: 5.5.5.0-018R).
Factor-TS Dionis NX	[OOTB] Factor-TS Dionis NX syslog	regexp	Designed for processing some audit events received from the Dionis-NX system (version 2.0.3) via Syslog.
F5 Advanced Web Application Firewall	[OOTB] F5 Advanced Web Application Firewall syslog	regexp	Designed for processing audit events received from the F5 Advanced Web Application Firewall system via Syslog.
F5 BigIP Advanced Firewall Manager (AFM)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FFRI FFR yarai	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FireEye CM Series	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FireEye Malware Protection System	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Forcepoint NGFW	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Forcepoint SMC	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Fortinet FortiAnalyzer	[OOTB] Syslog-CEF	Syslog	Designed for processing events received from Fortinet FortiAnalyzer 7.0, 7.2 via Syslog in CEF format.
Fortinet FortiGate	[OOTB] Syslog-CEF	regexp	Designed for processing events received from Fortinet FortiGate 7.0, 7.2 via Syslog in CEF format.
Fortinet FortiGate	[OOTB] FortiGate syslog KV	Syslog	Designed for processing events from FortiGate firewalls (version 7.0) via Syslog. The event source is FortiGate logs in key-value format.
Fortinet Fortimail	[OOTB] Fortimail	regexp	Designed for processing events of the FortiMail email protection system. The event source is Fortimail mail system logs.
Fortinet FortiSOAR	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
FreeBSD	[OOTB] FreeBSD file	regexp	Designed for processing events of the FreeBSD operating system (version 13.1-RELEASE) stored in a file. The normalizer can process files produced by the praudit utility. Example: praudit -xl /var/audit/AUDITFILE >> file_name.log

System name	Normalizer name	Type	Normalizer description
FreeIPA	[OOTB] FreeIPA	json	Designed for processing events from the FreeIPA system. The event source is Free IPA directory service logs.
FreeRADIUS	[OOTB] FreeRADIUS syslog	Syslog	Designed for processing events of the FreeRADIUS system received via Syslog. The normalizer supports events from FreeRADIUS version 3.0.
GajShield Firewall	[OOTB] GajShield Firewall syslog	regexp	Designed for processing part of the events received from the GajShield Firewall version GAJ_OS_Bulwark_Firmware_v4.35 via Syslog.
Garda Monitor	[OOTB] Garda Monitor syslog	Syslog	Designed for processing events of the Garda Monitor system version 3.4 received via Syslog.
Gardatech GardaDB	[OOTB] Gardatech GardaDB syslog	Syslog	Designed for processing events of the Gardatech Perimeter system version 5.3, 5.4 received via Syslog.
Gardatech Perimeter	[OOTB] Gardatech Perimeter syslog	Syslog	Designed for processing events of the Gardatech Perimeter system version 5.3 received via Syslog.
Gigamon GigaVUE	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
HAProxy	[OOTB] HAProxy syslog	Syslog	Designed for processing logs of the HAProxy system. The normalizer supports events of the HTTP log, TCP log, Error log type from HAProxy version 2.8.
HashiCorp Vault	[OOTB] HashiCorp Vault json	json	Designed for processing events received from the HashiCorp Vault system version 1.16 in JSON format. The normalizer package is available in KUMA 3.0 and later.
Huawei Eudemon	[OOTB] Huawei Eudemon	regexp	Designed for processing events from Huawei Eudemon firewalls. The event source is logs of Huawei Eudemon firewalls.
Huawei iManager 2000	[OOTB] Huawei iManager 2000 file	regexp	This normalizer supports processing some of the events of the Huawei iManager 2000 system, which are stored in the \client\logs\rpc, \client\logs\deploy\ossDeployment files.
Huawei USG	[OOTB] Huawei USG Basic	Syslog	Designed for processing events received from Huawei USG security gateways via Syslog.
Huawei VRP	[OOTB] Huawei VRP syslog	regexp	Designed for processing some types of Huawei VRP system events received via Syslog. The normalizer makes a partial selection of event data. The normalizer is available in KUMA 3.0 and later.
IBM InfoSphere Guardium	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Ideco UTM	[OOTB] Ideco UTM Syslog	Syslog	Designed for processing events received from Ideco UTM via Syslog. The normalizer supports events of Ideco UTM 14.7, 14.10, 17.5.
Illumio Policy Compute Engine (PCE)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Imperva Incapsula	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Imperva SecureSphere	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Indeed Access Manager	[OOTB] Indeed Access Manager syslog	Syslog	Designed for processing events received from the Indeed Access Manager system via Syslog.
Indeed PAM	[OOTB] Indeed PAM syslog	Syslog	Designed for processing events of Indeed PAM (Privileged Access Manager) version 2.6.
Indeed SSO	[OOTB] Indeed SSO xml	xml	Designed for processing events of the Indeed SSO (Single Sign-On) system. The normalizer supports KUMA 2.1.3 and later.
InfoWatch Person Monitor	[OOTB] InfoWatch Person Monitor SQL	sql	Designed for processing system audit events from the MS SQL database of InfoWatch Person Monitor 10.2.
InfoWatch Traffic Monitor	[OOTB] InfoWatch Traffic Monitor SQL	sql	Designed for processing events received by the connector from the database of the InfoWatch Traffic Monitor system.

System name	Normalizer name	Type	Normalizer description
Intralinks VIA	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
IPFIX	[OOTB] IPFIX	ipfix	Designed for processing events in the IP Flow Information Export (IPFIX) format.
Juniper JUNOS	[OOTB] Juniper - JUNOS	regexp	Designed for processing audit events received from Juniper network devices.
Kaspersky Anti Targeted Attack (KATA)	[OOTB] KATA	cef	Designed for processing alerts or events from the Kaspersky Anti Targeted Attack activity log.
Kaspersky CyberTrace	[OOTB] CyberTrace	regexp	Designed for processing Kaspersky CyberTrace events.
Kaspersky Endpoint Detection and Response (KEDR)	[OOTB] KEDR telemetry	json	Designed for processing Kaspersky EDR telemetry tagged by KATA. The event source is kafka, EnrichedEventTopic
Kaspersky Endpoint Security for Linux	[OOTB] KESL syslog cef	Syslog	Designed for processing events from Kaspersky Endpoint Security for Linux (KESL) 12.2 in CEF format via Syslog.
KICS/KATA	[OOTB] KICS4Net v2.x	cef	Designed for processing KICS/KATA version 2.x events.
KICS/KATA	[OOTB] KICS4Net v3.x	Syslog	Designed for processing KICS/KATA version 3.x events.
KICS/KATA 4.2	[OOTB] Kaspersky Industrial CyberSecurity for Networks 4.2 syslog	Syslog	Designed for processing events received from the KICS/KATA 4.2 system via Syslog.
Kaspersky KISG	[OOTB] Kaspersky KISG syslog	Syslog	Designed for processing events received from Kaspersky IoT Secure Gateway (KISG) 3.0 via Syslog.
Open Single Management Platform	[OOTB] KSC	cef	Designed for processing Open Single Management Platform events received in CEF format.
Open Single Management Platform	[OOTB] KSC from SQL	sql	Designed for processing events received by the connector from the database of the Open Single Management Platform system.
Kaspersky Security for Linux Mail Server (KLMS)	[OOTB] KLMS Syslog CEF	Syslog	Designed for processing events from Kaspersky Security for Linux Mail Server in CEF format via Syslog.
Kaspersky Security for MS Exchange SQL	[OOTB] Kaspersky Security for MS Exchange SQL	sql	Normalizer for Kaspersky Security for Exchange (KSE) 9.0 events stored in the database.
Kaspersky Secure Mail Gateway (KSMG)	[OOTB] KSMG Syslog CEF	Syslog	Designed for processing events of Kaspersky Secure Mail Gateway version 2.0 in CEF format via Syslog.
Kaspersky Web Traffic Security (KWTS)	[OOTB] KWTS Syslog CEF	Syslog	Designed for processing events received from Kaspersky Web Traffic Security in CEF format via Syslog.
Kaspersky Web Traffic Security (KWTS)	[OOTB] KWTS (KV)	Syslog	Designed for processing events in Kaspersky Web Traffic Security for Key-Value format.
Kemptechnologies LoadMaster	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Kerio Control	[OOTB] Kerio Control	Syslog	Designed for processing events of Kerio Control firewalls.

System name	Normalizer name	Type	Normalizer description
KUMA	[OOTB] KUMA forwarding	json	Designed for processing events forwarded from KUMA.
LastLine Enterprise	[OOTB] LastLine Enterprise syslog cef	Syslog	Designed for processing events received from LastLine Enterprise 7.3, 8.1, 9.1 via Syslog in CEF format.
Libvirt	[OOTB] Libvirt syslog	Syslog	Designed for processing events of Libvirt version 8.0.0 received via Syslog.
Lieberman Software ERPM	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Linux	[OOTB] Linux audit and iptables Syslog v1	Syslog	Designed for processing events of the Linux operating system. This normalizer does not support processing events in the ENRICHED format.
Linux auditd	[OOTB] Linux auditd syslog for KUMA 3.2	Syslog	Designed for processing audit events (auditd package) of the Linux operating system received via Syslog. The normalizer supports events that have been processed by a KUMA collector version 3.2 or later.
Linux auditd	[OOTB] Linux auditd file for KUMA 3.2	regexp	Designed for processing audit events (auditd package) of the Linux operating system saved to a file. The normalizer supports events that have been processed by a KUMA collector version 3.2 or later.
MariaDB	[OOTB] MariaDB Audit Plugin Syslog	Syslog	Designed for processing events coming from the MariaDB audit plugin over Syslog.
McAfee Endpoint DLP	[OOTB] McAfee Endpoint DLP syslog	Syslog	Designed for processing events received from McAfee Endpoint DLP Windows 11.10.200 via Syslog.
Microsoft 365 (Office 365)	[OOTB] Microsoft Office 365 json	json	This normalizer is designed for processing Microsoft 365 events.
Microsoft Active Directory Federation Service (AD FS)	[OOTB] Microsoft Products for KUMA 3	xml	Designed for processing Microsoft AD FS events. The [OOTB] Microsoft Products for KUMA 3 normalizer supports this event source in KUMA 3.0.1 and later versions.
Microsoft Active Directory Domain Service (AD DS)	[OOTB] Microsoft Products for KUMA 3	xml	Designed for processing Microsoft AD DS events. The [OOTB] Microsoft Products for KUMA 3 normalizer supports this event source in KUMA 3.0.1 and later versions.
Microsoft Defender	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Defender events.
Microsoft DHCP	[OOTB] MS DHCP file	regexp	Designed for processing Microsoft DHCP server events. The event source is Windows DHCP server logs.
Microsoft DNS	[OOTB] DNS Windows	regexp	Designed for processing Microsoft DNS server events. The event source is Windows DNS server logs. The normalizer does not support processing debug log events with the **Details** option enabled.
Microsoft Exchange	[OOTB] Exchange CSV	csv	Designed for processing the event log of the Microsoft Exchange system. The event source is Exchange server MTA logs.
Microsoft Hyper-V	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows events. The event source is Microsoft Hyper-V logs: Microsoft-Windows-Hyper-V-VMMS-Admin, Microsoft-Windows-Hyper-V-Compute-Operational, Microsoft-Windows-Hyper-V-Hypervisor-Operational, Microsoft-Windows-Hyper-V-StorageVSP-Admin, Microsoft-Windows-Hyper-V-Hypervisor-Admin, Microsoft-Windows-Hyper-V-VMMS-Operational, Microsoft-Windows-Hyper-V-Compute-Admin.
Microsoft IIS	[OOTB] IIS Log File Format	regexp	The normalizer processes events in the format described at https://learn.microsoft.com/en-us/windows/win32/http/iis-logging. The event source is Microsoft IIS logs.

System name	Normalizer name	Type	Normalizer description
Microsoft Network Policy Server (NPS)	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	The normalizer is designed for processing events of the Microsoft Windows operating system. The event source is Network Policy Server events.
Microsoft SCCM	[OOTB] Microsoft SCCM file	regexp	Designed for processing events of the Microsoft SCCM system version 2309. The normalizer supports processing of some of the events stored in the AdminService.log file.
Microsoft SharePoint Server	[OOTB] Microsoft SharePoint Server diagnostic log file	regexp	The normalizer supports processing part of Microsoft SharePoint Server 2016 events stored in diagnostic logs.
Microsoft Sysmon	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	This normalizer is designed for processing Microsoft Sysmon module events.
Microsoft Windows 7, 8.1, 10, 11	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing part of events from the Security, System, Application logs of the Microsoft Windows operating system. The [OOTB] Microsoft Products via KES WIN normalizer supports a limited number of audit event types sent to KUMA by Kaspersky Endpoint Security 12.6 for Windows via Syslog.
Microsoft-Windows-PowerShell	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows PowerShell log events.
Microsoft-Windows-PowerShell-Operational	[OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows PowerShell-Operational log events. The [OOTB] Microsoft Products via KES WIN normalizer supports a limited number of audit event types sent to KUMA by Kaspersky Endpoint Security 12.6 for Windows via Syslog.
Microsoft SQL Server	[Deprecated] [OOTB] Microsoft SQL Server xml	xml	Designed for processing events of MS SQL Server versions 2008, 2012, 2014, 2016. The normalizer supports KUMA 2.1.3 and later.
Microsoft Windows Remote Desktop Services	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing Microsoft Windows events. The event source is the log at Applications and Services Logs - Microsoft - Windows - TerminalServices-LocalSessionManager - Operational The [OOTB] Microsoft Products via KES WIN normalizer supports a limited number of audit event types sent to KUMA by Kaspersky Endpoint Security 12.6 for Windows via Syslog.
Microsoft Windows Server 2008 R2, 2012 R2, 2016, 2019, 2022	[OOTB] Microsoft Products, [OOTB] Microsoft Products for KUMA 3, [OOTB] Microsoft Products via KES WIN	xml	Designed for processing part of events from the Security, System logs of the Microsoft Windows Server operating system. The [OOTB] Microsoft Products via KES WIN normalizer supports a limited number of audit event types sent to KUMA by Kaspersky Endpoint Security 12.6 for Windows via Syslog.
Microsoft Windows XP/2003	[OOTB] SNMP. Windows {XP/2003}	json	Designed for processing events received from workstations and servers running Microsoft Windows XP, Microsoft Windows 2003 operating systems using the SNMP protocol.

System name	Normalizer name	Type	Normalizer description
Microsoft WSUS	[OOTB] Microsoft WSUS file	regexp	Designed for processing events of the Gardatech Perimeter system version 5.3, 5.4 received via Syslog.
MikroTik	[OOTB] MikroTik syslog	regexp	Designed for events received from MikroTik devices via Syslog.
Minerva Labs Minerva EDR	[OOTB] Minerva EDR	regexp	Designed for processing events from the Minerva EDR system.
MongoDb	[OOTB] MongoDb syslog	Syslog	Designed for processing part of events received from the MongoDB 7.0 database via Syslog.
Multifactor Radius Server for Windows	[OOTB] Multifactor Radius Server for Windows syslog	Syslog	Designed for processing events received from the Multifactor Radius Server 1.0.2 for Microsoft Windows via Syslog.
MySQL 5.7	[OOTB] MariaDB Audit Plugin Syslog	Syslog	Designed for processing events coming from the MariaDB audit plugin over Syslog.
NetApp ONTAP (AFF, FAM)	[OOTB] NetApp syslog, [OOTB] NetApp file	regexp	[OOTB] NetApp syslog — designed for processing events of the NetApp system (version — ONTAP 9.12) received via Syslog. [OOTB] NetApp file — designed for processing events of the NetApp system (version — ONTAP 9.12) stored in a file.
NetApp SnapCenter	[OOTB] NetApp SnapCenter file	regexp	Designed to process part of the events of the NetApp SnapCenter system (SnapCenter Server 5.0). The normalizer supports processing some of the events from the C:\Program Files\NetApp\SnapCenter WebApp\App_Data\log\napManagerWeb.*.log file. Types of supported events in xml format from the SnapManagerWeb.*.log file: SmDiscoverPluginRequest, SmDiscoverPluginResponse, SmGetDomainsResponse, SmGetHostPluginStatusRequest, SmGetHostPluginStatusResponse, SmGetHostRequest, SmGetHostResponse, SmRequest. The normalizer supports processing some of the events from the C:\Program Files\NetApp\SnapCenter WebApp\App_Data\log\audit.log file.
NetIQ Identity Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
NetScout Systems nGenius Performance Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Netskope Cloud Access Security Broker	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Netwrix Auditor	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Nextcloud	[OOTB] Nextcloud syslog	Syslog	Designed for events of Nextcloud version 26.0.4 received via Syslog. The normalizer does not save information from the Trace field.
Nexthink Engine	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Nginx	[OOTB] Nginx regexp	regexp	Designed for processing Nginx web server log events.
NIKSUN NetDetector	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
One Identity Privileged Session Management	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
OpenLDAP	[OOTB] OpenLDAP	regexp	Designed for line-by-line processing of some events of the OpenLDAP 2.5 system in an auditlog.ldif file.
Open VPN	[OOTB] OpenVPN file	regexp	Designed for processing the event log of the OpenVPN system.
Oracle	[OOTB] Oracle Audit Trail	sql	Designed for processing database audit events received by the connector directly from an Oracle database.

System name	Normalizer name	Type	Normalizer description
OrionSoft Termit	[OOTB] OrionSoft Termit syslog	Syslog	Designed for processing events received from the OrionSoft Termit 2.2 system via Syslog.
Orion soft zVirt	[OOTB] Orion Soft zVirt syslog	regexp	Designed for processing events of the Orion soft zVirt 3.1 virtualization system.
PagerDuty	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Palo Alto Cortex Data Lake	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Palo Alto Networks NGFW	[OOTB] PA-NGFW (Syslog-CSV)	Syslog	Designed for processing events from Palo Alto Networks firewalls received via Syslog in CSV format.
Palo Alto Networks PANOS	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Parsec ParsecNet	[OOTB] Parsec ParsecNet	sql	Designed for processing events received by the connector from the database of the Parsec ParsecNet 3 system.
Passwork	[OOTB] Passwork syslog	Syslog	Designed for processing events received from the Passwork version 050219 system via Syslog.
Penta Security WAPPLES	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Positive Technologies ISIM	[OOTB] PTsecurity ISIM	regexp	Designed for processing events from the PT Industrial Security Incident Manager system.
Positive Technologies Network Attack Discovery (NAD)	[OOTB] PT NAD json	json	Designed for processing events coming from PT NAD in json format. This normalizer supports events from PT NAD version 11.1, 11.0.
Positive Technologies Sandbox	[OOTB] PTsecurity Sandbox	regexp	Designed for processing events of the PT Sandbox system.
Positive Technologies Web Application Firewall	[OOTB] PTsecurity WAF	Syslog	Designed for processing events from the PTsecurity (Web Application Firewall) system.
Postfix	[OOTB] Postfix syslog	regexp	The [OOTB] Postfix package contains a resource set for processing Postfix 3.6 events. It supports processing Syslog events received over TCP. The package is available for KUMA 3.0 and newer versions.
PostgreSQL pgAudit	[OOTB] PostgreSQL pgAudit Syslog	Syslog	Designed for processing events of the pgAudit audit plug-in for the PostgreSQL database received via Syslog.
PowerDNS	[OOTB] PowerDNS syslog	Syslog	Designed for processing events of PowerDNS Authoritative Server 4.5 received via Syslog.
Proftpd	[OOTB] Proftpd syslog	regexp	Designed for processing events received from Proftpd 1.3.8c via Syslog.
Proofpoint Insider Threat Management	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Proxmox	[OOTB] Proxmox file	regexp	Designed for processing events of the Proxmox system version 7.2-3 stored in a file. The normalizer supports processing of events in access and pveam logs.
PT NAD	[OOTB] PT NAD json	json	Designed for processing events coming from PT NAD in json format. This normalizer supports events from PT NAD version 11.1, 11.0.
QEMU - hypervisor logs	[OOTB] QEMU - Hypervisor file	regexp	Designed for processing events of the QEMU hypervisor stored in a file. QEMU 6.2.0 and Libvirt 8.0.0 are supported.
QEMU - virtual machine logs	[OOTB] QEMU - Virtual Machine file	regexp	Designed for processing events from logs of virtual machines of the QEMU hypervisor version 6.2.0, stored in a file.
Radware DefensePro AntiDDoS	[OOTB] Radware DefensePro AntiDDoS	Syslog	Designed for processing events from the DDOS Mitigator protection system received via Syslog.

System name	Normalizer name	Type	Normalizer description
Reak Soft Blitz Identity Provider	[OOTB] Reak Soft Blitz Identity Provider file	regexp	Designed for processing events of the Reak Soft Blitz Identity Provider system version 5.16, stored in a file.
RedCheck Desktop	[OOTB] RedCheck Desktop file	regexp	Designed for processing logs of the RedCheck Desktop 2.6 system stored in a file.
RedCheck WEB	[OOTB] RedCheck WEB file	regexp	Designed for processing logs of the RedCheck Web 2.6 system stored in files.
RED SOFT RED ADM	[OOTB] RED SOFT RED ADM syslog	regexp	Designed for processing events received from the RED ADM system (RED ADM: Industrial edition 1.1) via syslog. The normalizer supports processing: - Management subsystem events - Controller events
ReversingLabs N1000 Appliance	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Rubicon Communications pfSense	[OOTB] pfSense Syslog	Syslog	Designed for processing events from the pfSense firewall received via Syslog.
Rubicon Communications pfSense	[OOTB] pfSense w/o hostname	Syslog	Designed for processing events from the pfSense firewall. The Syslog header of these events does not contain a hostname.
SailPoint IdentityIQ	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
SecurityCode Continent 4	[OOTB] SecurityCode Continent 4 syslog	regexp	Designed for processing events of the SecurityCode Continent system version 4 received via Syslog.
Sendmail	[OOTB] Sendmail syslog	Syslog	Designed for processing events of Sendmail version 8.15.2 received via Syslog.
SentinelOne	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Skype for Business	[OOTB] Microsoft Products for KUMA 3	xml	Designed for processing some of the events from the log of the Skype for Business system, the Lync Server log.
Snort	[OOTB] Snort 3 json file	json	Designed for processing events of Snort version 3 in JSON format.
Sophos Central	[OOTB] Sophos Central syslog	Syslog	Designed for processing some events received from Sophos Central 1.2 via Syslog in CEF format from the Sophos-Central-SIEM-Integration script.
Sonicwall TZ	[OOTB] Sonicwall TZ Firewall	Syslog	Designed for processing events received via Syslog from the SonicWall TZ firewall.
Solar WebProxy	[OOTB] Solar WebProxy syslog	regexp	Designed for processing events received from Solar WebProxy 4.2 in siem-log format via Syslog.
SolarWinds DameWare MRC	[OOTB] SolarWinds DameWare MRC xml	xml	This normalizer supports processing some of the DameWare Mini Remote Control (MRC) 7.5 events stored in the Application log of Windows. The normalizer processes events generated by the dwmrcs provider.
Sophos Firewall	[OOTB] Sophos Firewall syslog	regexp	Designed for processing events received from Sophos Firewall 20 via Syslog.
Sophos XG	[OOTB] Sophos XG	regexp	Designed for processing events from the Sophos XG firewall.
Squid	[OOTB] Squid access Syslog	Syslog	Designed for processing events of the Squid proxy server received via the Syslog protocol.
Squid	[OOTB] Squid access.log file	regexp	Designed for processing Squid log events from the Squid proxy server. The event source is access.log logs

System name	Normalizer name	Type	Normalizer description
Staffcop Enterprise	[OOTB] Staffcop Enterprise syslog CEF	regexp	Designed for processing events received from Staffcop Enterprise 5.4, 5.5 in CEF format via Syslog.
S-Terra VPN Gate	[OOTB] S-Terra	Syslog	Designed for processing events from S-Terra VPN Gate devices.
Suricata	[OOTB] Suricata json file	json	This package contains a normalizer for Suricata 7.0.1 events stored in a JSON file. The normalizer supports processing the following event types: flow, anomaly, alert, dns, http, ssl, tls, ftp, ftp_data, ftp, smb, rdp, pgsql, modbus, quic, dhcp, bittorrent_dht, rfb.
ThreatConnect Threat Intelligence Platform	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
ThreatQuotient	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Tionix Cloud Platform	[OOTB] Tionix Cloud Platform syslog	Syslog	Designed for processing events of the Tionix Cloud Platform system version 2.9 received via Syslog. The normalizer makes a partial selection of event data. The normalizer is available in KUMA 3.0 and later.
Tionix VDI	[OOTB] Tionix VDI file	regexp	This normalizer supports processing some of the Tionix VDI system (version 2.8) events stored in the tionix_lntmov.log file.
TrapX DeceptionGrid	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trend Micro Control Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trend Micro Deep Security	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trend Micro NGFW	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Trustwave Application Security DbProtect	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Unbound	[OOTB] Unbound Syslog	Syslog	Designed for processing events from the Unbound DNS server received via Syslog.
UserGate	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the UserGate system via Syslog.
Varonis DatAdvantage	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Veriato 360	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
ViPNet TIAS	[OOTB] Vipnet TIAS syslog	Syslog	Designed for processing events of ViPNet TIAS 3.8 received via Syslog.
VMware ESXi	[OOTB] VMware ESXi syslog	regexp	Designed for processing VMware ESXi events (support for a limited number of events from ESXi versions 5.5, 6.0, 6.5, 7.0) received via Syslog.
VMWare Horizon	[OOTB] VMware Horizon - Syslog	Syslog	Designed for processing events received from the VMware Horizon 2106 system via Syslog.
VMwareCarbon Black EDR	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Vmware vCenter	[OOTB] VMware vCenter API	xml	Designed for processing VMware vCenter 7 events received via API.
Vormetric Data Security Manager	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Votiro Disarmer for Windows	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Vsftpd	[OOTB] Vsftpd syslog	regexp	Designed for processing events received from Vsftpd 3.0.5 via Syslog.

System name	Normalizer name	Type	Normalizer description
Wallix AdminBastion	[OOTB] Wallix AdminBastion syslog	regexp	Designed for processing events received from the Wallix AdminBastion system via Syslog.
WatchGuard - Firebox	[OOTB] WatchGuard Firebox	Syslog	Designed for processing WatchGuard Firebox events received via Syslog.
Webroot BrightCloud	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Winchill Fracas	[OOTB] PTC Winchill Fracas	regexp	Designed for processing events of the Windchill FRACAS failure registration system.
Yandex Browser corporate	[OOTB] Yandex Browser	json	Designed for processing events received from the corporate version of Yandex Browser 23 or 24.4.
Yandex Cloud	[OOTB] Yandex Cloud	regexp	Designed for processing part of Yandex Cloud audit events. The normalizer supports processing audit log events of the configuration level: IAM (Yandex Identity and Access Management), Compute (Yandex Compute Cloud), Network (Yandex Virtual Private Cloud), Storage (Yandex Object Storage), Resourcemanager (Yandex Resource Manager).
Zabbix	[OOTB] Zabbix SQL	sql	Designed for processing events of Zabbix 6.4.
Zecurion DLP	[OOTB] Zecurion DLP syslog	regexp	Designed for processing events of the Zecurion DLP system version 12.0 received via Syslog.
ZEEK IDS	[OOTB] ZEEK IDS json file	json	Designed for processing logs of the ZEEK IDS system in JSON format. The normalizer supports events from ZEEK IDS version 1.8.
Zettaset BDEncrypt	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
Zscaler Nanolog Streaming Service (NSS)	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format.
IT-Bastion – SKDPU	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the IT-Bastion SKDPU system via Syslog.
A-Real Internet Control Server (ICS)	[OOTB] A-real IKS syslog	regexp	Designed for processing events of the A-Real Internet Control Server (ICS) system received via Syslog. The normalizer supports events from A-Real ICS version 7.0 and later.
Apache web server	[OOTB] Apache HTTP Server file	regexp	Designed for processing Apache HTTP Server 2.4 events stored in a file. The normalizer supports processing of events from the Application log in the Common or Combined Log formats, as well as the Error log. Expected format of the Error log events: `[%t] [%-m:%l] [pid %P:tid %T] [server\ %v] [client\ %a] %E: %M;\ referer\ %-{Referer}i`
Apache web server	[OOTB] Apache HTTP Server syslog	Syslog	Designed for processing events of the Apache HTTP Server received via Syslog. The normalizer supports processing of Apache HTTP Server 2.4 events from the Access log in the Common or Combined Log format, as well as the Error log. Expected format of the Error log events: `[%t] [%-m:%l] [pid %P:tid %T] [server\ %v] [client\ %a] %E: %M;\ referer\ %-{Referer}i`
Lighttpd web server	[OOTB] Lighttpd syslog	Syslog	Designed for processing Access events of the Lighttpd system received via Syslog. The normalizer supports processing of Lighttpd version 1.4 events. Expected format of Access log events: `$remote_addr $http_request_host_name $remote_user [$time_local] "$request" $status $body_bytes_sent "$http_referer" "$http_user_agent"`
IVK Kolchuga-K	[OOTB] Kolchuga-K Syslog	Syslog	Designed for processing events from the IVK Kolchuga-K system, version LKNV.466217.002, via Syslog.
infotecs ViPNet IDS	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the infotecs ViPNet IDS system via Syslog.
infotecs ViPNet Coordinator	[OOTB] VipNet Coordinator Syslog	Syslog	Designed for processing events from the ViPNet Coordinator system received via Syslog.

System name	Normalizer name	Type	Normalizer description
Kod Bezopasnosti — Continent	[OOTB][regexp] Continent IPS/IDS & TLS	regexp	Designed for processing events of Continent IPS/IDS device log.
Kod Bezopasnosti — Continent	[OOTB] Continent SQL	sql	Designed for getting events of the Continent system from the database.
Kod Bezopasnosti SecretNet 7	[OOTB] SecretNet SQL	sql	Designed for processing events received by the connector from the database of the SecretNet system.
Confident – Dallas Lock Unified Control Center	[OOTB] Confident Dallas Lock syslog CEF	regexp	Designed for processing events received from Dallas Lock Unified Control Center 4.0 in CEF format.
CryptoPro NGate	[OOTB] Ngate Syslog	Syslog	Designed for processing events received from the CryptoPro NGate system via Syslog.
H3C (Huawei-3Com) routers	[OOTB] H3C Routers syslog	regexp	Normalizer for some types of events received from H3C (Huawei-3Com) SR6600 network devices (Comware 7 firmware) via Syslog. The normalizer supports the standard event format (RFC 3164-compliant format).
NT Monitoring and Analytics	[OOTB] Syslog-CEF	Syslog	Designed for processing events in the CEF format received from the NT Monitoring and Analytics system via Syslog.
BlueCoat proxy server	[OOTB] BlueCoat Proxy v0.2	regexp	Designed to process BlueCoat proxy server events. The event source is the BlueCoat proxy server event log.
SKDPU NT Access Gateway	[OOTB] Bastion SKDPU-GW	Syslog	Designed for processing events of the SKDPU NT Access gateway system received via Syslog.
Solar Dozor	[OOTB] Solar Dozor Syslog	Syslog	Designed for processing events received from the Solar Dozor system version 7.9 via Syslog. The normalizer supports custom format events and does not support CEF format events.
-	[OOTB] Syslog header	Syslog	Designed for processing events received via Syslog. The normalizer parses the header of the Syslog event, the message field of the event is not parsed. If necessary, you can parse the message field using other normalizers.

Aggregation rules

Aggregation rules let you combine repetitive events of the same type and replace them with one common event. Aggregation rules support fields of the standard KUMA event schema as well as fields of the extended event schema. In this way, you can reduce the number of similar events sent to the storage and/or correlator, reduce the workload on services, conserve data storage space and licensing quota (EPS). An aggregation event is created when a time or number of events threshold is reached, whichever occurs first.

For aggregation rules, you can configure a filter and apply it only to events that match the specified conditions.

You can configure aggregation rules under **Monitoring** → **Resources** → **Aggregation rules**, and then select the created aggregation rule from the drop-down list in the collector settings. You can also configure aggregation rules directly in the collector settings.

Available aggregation rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Threshold	Threshold on the number of events. After accumulating the specified number of events with identical fields, the collector creates an aggregation event and begins accumulating events for the next aggregated event. The default value is `100`.

Setting	Description
Triggered rule lifetime	Required setting. Threshold on time in seconds. When the specified time expires, the accumulation of base events stops, the collector creates an aggregated event and starts obtaining events for the next aggregated event. The default value is `60`.
Description	Resource description: up to 4,000 Unicode characters.
Identical fields	Required setting. This drop-down list lists the fields of normalized events that must have identical values. For example, for network events, you can use SourceAddress, DestinationAddress, DestinationPort fields. In the aggregation event, these fields are populated with the values of the base events.
Unique fields	This drop-down list lists the fields whose range of values must be saved in the aggregated event. For example, if the DestinationPort field is specified under **Unique fields** and not **Identical fields**, the aggregated event combines base connection events for a variety of ports, and the DestinationPort field of the aggregated event contains a list of all ports to which connections were made.
Sum fields	In this drop-down list, you can select the fields whose values will be summed up during aggregation and written to the same-name fields of the aggregated event.

Setting	Description
Filter	Group of settings in which you can specify the conditions for identifying events that must be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter. In aggregation rules, do not use filters with the **TI** operand or the **TIDetect**, **inActiveDirectoryGroup**, or **hasVulnerability** operators. The Active Directory fields for which you can use the **inActiveDirectoryGroup** operator will appear during the enrichment stage (after aggregation rules are executed). **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when you edit the settings of the created aggregation rule; on this tab, you can link the rule to a collector.

To link an aggregation rule to a collector, click the **Link to service** button → **Link to collector**. This opens a window; in that window, select one or more collectors and click **OK**. The aggregation rule is linked to the selected collectors and is applied in accordance with their configuration.

If necessary, you can unlink the aggregation rule from a collector. To do this, select the check box next to the collector, click **Delete** and confirm the unlinking from the service.

Predefined aggregation rules

The OSMP distribution kit includes aggregation rules listed in the table below.

Aggregation rule name	Description
[OOTB] Netflow 9	The rule is triggered after 100 events or 10 seconds. Events are aggregated by fields: • DestinationAddress • DestinationPort • SourceAddress • TransportProtocol • DeviceVendor • DeviceProduct The DeviceCustomString1 and BytesIn fields are summed up.

Enrichment rules

Event enrichment involves adding information to events that can be used to identify and investigate an incident.

Enrichment rules let you add supplementary information to event fields by transforming data that is already present in the fields, or by querying data from external systems. For example, suppose that a user name is recorded in the event. You can use an enrichment rule to add information about the department, position, and manager of this user to the event fields.

Enrichment rules can be used in the following KUMA services and features:

- Collector.
- Correlator.
- Normalizer.

Available enrichment rule settings are listed in the table below.

Basic settings tab

Setting	Description

Source kind	Required setting. Drop-down list for selecting the type of incoming events. Depending on the selected type, you may see the following additional settings:

- **constant**

  This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

  If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

  If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- **dictionary**

  This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

  If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

  Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

  If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the " | " character when the dictionary is queried.

  Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **table**

  This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

  When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, click the **Add field** button to select the event fields whose values are used for dictionary entry selection.

  In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

  - In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
  - In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

  New table rows can be added by clicking the **Add new element** button. Columns can be deleted by clicking the ✕ button.

- **event**

  This type of enrichment is used when you need to write a value from another event field to the current event field. Settings of this type of enrichment:

  - In the **Target** field drop-down list, select the KUMA event field to which you want to write the data.
  - In the **Source** field drop-down list, select the event field whose value will be written to the target field.
  - In the **Conversion settings** block, you can create rules for modifying the original data before it is written to the KUMA event fields. The conversion type can be selected from the drop-down list. You can click the **Add conversion** and **Delete** buttons to add or delete a conversion, respectively. The order of conversions is important.

Available conversions

Conversions are changes that can be applied to a value before it gets written to the event field. The conversion type is selected from a drop-down list.

Available conversions:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and this password gets logged in plain text.
- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using the regular expression RE2. When this conversion type is selected, the field appears where regular expression should be added.
- **substring**—is used to extract characters in the position range specified in the **Start** and **End** fields. These fields appear when this conversion type is selected.
- **replace**—is used to replace specified character sequence with the other character sequence. When this type of conversion is selected, new fields appear:
  - **Replace chars**—in this field you can specify the character sequence that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- **trim**—used to simultaneously remove the characters specified in the Chars field from the leading and end positions of the value. The field appears when this type of conversion is selected. For example, a trim conversion with the Micromon value applied to Microsoft-Windows-Sysmon results in soft-Windows-Sys.
- **append** is used to add the characters specified in the Constant field to the end of the event field value. The field appears when this type of conversion is selected.
- **prepend**—used to prepend the characters specified in the Constant field to the start of the event field value. The field appears when this type of conversion is selected.
- **replace with regexp**—is used to replace RE2 regular expression results with the character sequence.
  - **Expression**—in this field you can specify the regular expression which results that should be replaced.
  - **With chars**—in this field you can specify the characters sequence should be used instead of replaced characters.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, such a string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: regexp, substring, replace, trim, append, prepend, replaceWithRegexp, decodeHexString, decodeBase64String, decodeBase64URLString.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: append, prepend.

- **template**

This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- `{{- range $index, $element := . SA.StringArrayOne -}}`
  `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString .SA.StringArray}}
```

- **dns**

This type of enrichment is used to send requests to a private network DNS server to convert IP addresses into domain names or vice versa. IP addresses are converted to DNS names only for private addresses: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 100.64.0.0/10.

Available settings:

- **URL**—in this field, you can specify the URL of a DNS server to which you want to send requests. You can use the **Add URL** button to specify multiple URLs.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Workers**—maximum number of requests per one point in time. The default value is `1`.
- **Max tasks**—maximum number of simultaneously fulfilled requests. By default, this value is equal to the number of vCPUs of the Open Single Management Platform Core server.
- **Cache TTL**—the lifetime of the values stored in the cache. The default value is `60`.
- **Cache disabled**—you can use this drop-down list to enable or disable caching. Caching is enabled by default.
- **The Recursion desired** is available starting with Open Single Management Platform 3.4.1. You can use this toggle switch to make an Open Single Management Platform collector send recursive queries to authoritative DNS servers for the purposes of enrichment. By default it is **Disabled**.

- **cybertrace**

This type of enrichment is deprecated, we recommend using cybertrace-http instead.

This type of enrichment is used to add information from CyberTrace data streams to event fields.

Available settings:

- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests. The default CyberTrace port is 9999.
- **Number of connections**—maximum number of connections to the CyberTrace server that can be simultaneously established by KUMA. By default, this value is equal to the number of vCPUs of the KUMA Core server.
- **RPS**—maximum number of requests sent to the server per second. The default value is `1,000`.
- **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is `30`.
- **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1,000,000,000.
- **Mapping** (required)—this settings block contains the mapping table for mapping KUMA event fields to CyberTrace indicator types. The **KUMA field** column shows the names of KUMA event fields, and the **CyberTrace indicator** column shows the types of CyberTrace indicators.

  Available types of CyberTrace indicators:

  - **ip**
  - **url**
  - **hash**

  In the mapping table, you must provide at least one string. You can use the **Add row** button to add a string, and can use the ✕ button to remove a string.

- **cybertrace-http**

This type of enrichment is used to add information from CyberTrace data streams to event fields using the REST API.

Available settings:

- **URL** (required)—in this field, you can specify the URL of a CyberTrace server to which you want to send requests.
- **Secret** (required) is a drop-down list in which you can select the secret which stores the credentials for the connection.
- **Timeout**—amount of time to wait for a response from the CyberTrace server, in seconds. The default value is 30.
- **Key fields** (required) is the list of event fields used for enriching events with data from CyberTrace.
- **Maximum number of events in the enrichment queue**—maximum number of events stored in the enrichment queue for re-sending. The default value is 1000000000. After reaching 1 million events received from the CyberTrace server, events stop being enriched until the number of received events is reduced to less than 500,000.

When the Queue figure reaches 1 million of received events, event enrichment stops, and events are recorded in the Storage unenriched until the Queue figure becomes less than 500 thousand events.

- **timezone**

This type of enrichment is used in collectors and correlators to assign a specific timezone to an event. Timezone information may be useful when searching for events that occurred at unusual times, such as nighttime.

When this type of enrichment is selected, the required timezone must be selected from the **Timezone** drop-down list.

Make sure that the required time zone is set on the server hosting the enrichment-utilizing service. For example, you can do this by using the `timedatectl list-timezones` command, which shows all time zones that are set on the server. For more details on setting time zones, please refer to your operating system documentation.

When an event is enriched, the time offset of the selected timezone relative to Coordinated Universal Time (UTC) is written to the DeviceTimeZone event field in the `+-hh:mm` format. For example, if you select the **Asia/Yekaterinburg** timezone, the value `+05:00` will be written to the DeviceTimeZone field. If the enriched event already has a value in the DeviceTimeZone field, it will be overwritten.

By default, if the timezone is not specified in the event being processed and enrichment rules by timezone are not configured, the event is assigned the timezone of the server hosting the service (collector or correlator) that processes the event. If the server time is changed, the service must be restarted.

Permissible time formats when enriching the DeviceTimeZone field

When processing incoming raw events in the collector, the following time formats can be automatically converted to the +-hh:mm format:

Time format in a processed event	Example
+-hh:mm	-07:00
+-hhmm	-0700
+-hh	-07

If the date format in the `DeviceTimeZone` field differs from the formats listed above, the collector server timezone is written to the field when an event is enriched with timezone information. You can create custom normalization rules for non-standard time formats.

- **geographic data**

This type of enrichment is used to add IP address geographic data to event fields. Learn more about linking IP addresses to geographic data.

When this type is selected, in the **Mapping geographic data to event fields** settings block, you must specify from which event field the IP address will be read, select the required attributes of geographic data, and define the event fields in which geographic data will be written:

1. In the **Event field with IP address** drop-down list, select the event field from which the IP address is read. Geographic data uploaded to KUMA is matched against this IP address.

   You can use the **Add event field with IP address** button to specify multiple event fields with IP addresses that require geographic data enrichment. You can delete event fields added in this way by clicking the **Delete event field with IP address** button.

   When the `SourceAddress`, `DestinationAddress`, and `DeviceAddress` event fields are selected, the **Apply default mapping** button becomes available. You can use this button to add preconfigured mapping pairs of geographic data attributes and event fields.

2. For each event field you need to read the IP address from, select the type of geographic data and the event field to which the geographic data should be written.

	You can use the **Add geodata attribute** button to add field pairs for **Geodata attribute** – **Event field to write to**. You can also configure different types of geographic data for one IP address to be written to different event fields. To delete a field pair, click ❌. • In the **Geodata attribute** field, select which geographic data corresponding to the read IP address should be written to the event. Available geographic data attributes: **Country**, **Region**, **City**, **Longitude**, **Latitude**. • In the **Event field to write to**, select the event field which the selected geographic data attribute must be written to. You can write identical geographic data attributes to different event fields. If you configure multiple geographic data attributes to be written to the same event field, the event will be enriched with the last mapping in the sequence.
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Debug	You can use this toggle switch to enable the logging of service operations. Logging is disabled by default.
Description	Resource description: up to 4,000 Unicode characters.

Filter	Group of settings in which you can specify the conditions for identifying events that must be processed by this resource. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when you edit the settings of the created enrichment rule; on this tab, you can link the rule to services, that is, correlators and collectors.

To link the enrichment rule to a collector or correlator, click the **Link to service** button → **Link to <service name>**. This opens a window; in that window, select one or more correlators or collectors and click **OK**. The enrichment rule is linked to the selected services and is applied in accordance with their configuration.

If necessary, you can unlink the aggregation rule from a service. To do this, select the check box next to the correlator or collector, click **Delete**, and confirm the unlinking from the service.

Predefined enrichment rules

The OSMP distribution kit includes enrichment rules listed in the table below.

Predefined enrichment rules

Enrichment rule name	Description
[OOTB] KATA alert	Used to enrich events received from KATA in the form of a hyperlink to an alert. The hyperlink is put in the DeviceExternalId field.

Correlation rules

Correlation rules are used to recognize specific sequences of processed events and to take certain actions after recognition, such as creating correlation events/alerts or interacting with an active list.

Correlation rules can be used in the following KUMA services and features:

- Correlator.
- Notification rule.
- Links of segmentation rules.
- Retroscan.

The available correlation rule settings depend on the selected type. Types of correlation rules:

- standard—used to find correlations between several events. Resources of this kind can create correlation events.

  This rule kind is used to determine complex correlation patterns. For simpler patterns you should use other correlation rule kinds that require less resources to operate.

- simple—used to create correlation events if a certain event is found.

- operational—used for operations with Active lists and context tables. This rule kind cannot create correlation events.

- periodical—used for detecting anomalous user activity based on accumulated data that is periodically analyzed. Anomalies are observed when the two metrics exceed thresholds: the **Relative Score** and the standard score (**Z-Score**).

  Rules of this type are available only when using the correlator-ng correlator. Such a rule cannot be created in the OSMP Console.

  When changing the correlator-ng configuration, you may need to delete the correlation/ and containers/ directories. Deleting these directories results in the loss of all accumulated correlator data, including statistics, correlation rule buckets, active lists, and context tables.

For these resources, you can enable the display of control characters in all input fields except the **Description** field.

If a correlation rule is used in the correlator and an alert was created based on it, any change to the correlation rule will not result in a change to the existing alert even if the correlator service is restarted. For example, if the name of a correlation rule is changed, the name of the alert will remain the same. If you close the existing alert, a new alert will be created and it will take into account the changes made to the correlation rule.

Standard correlation rules

Standard correlation rules are used to identify complex patterns in processed events.

The search for patterns is conducted by using buckets

Bucket is a data container that is used by the Correlation rule resources to determine if the correlation event should be created. It has the following functions:

- Group together events that were matched by the filters in the **Selectors** group of settings of the Correlation rule resource. Events are grouped by the fields that were selected by user in the **Identical fields** field.

- Determine the instance when the Correlation rule should trigger, affecting the events that are grouped in the bucket.

- Perform the actions that are selected in the **Actions** group of settings.

- Create correlation events.

Available states of the Bucket:

- `Empty`—the bucket has no events. This can happen only when it was created by the correlation rule triggering.

- `Partial Match`—the bucket has some of the expected events (recovery events are not counted).

- `Full Match`—the bucket has all of the expected events (recovery events are not counted). When this condition is achieved:
  - The Correlation rule triggers
  - Events are cleared from the bucket
  - The trigger counter of the bucket is updated
  - The state of the bucket becomes `Empty`
- `False Match`—this state of the Bucket is possible:
  - when the `Full Match` state was achieved but the join-filter returned false.
  - when **Recovery** check box was selected and the recovery events were received.

  When this condition is achieved the Correlation rule does not trigger. Events are cleared from the bucket, the trigger counter is updated, and the state of the bucket becomes `Empty`.

The correlation rule window contains the following tabs:

- **General**—used to specify the main settings of the correlation rule. On this tab, you can select the type of correlation rule.
- **Selectors**—used to define the conditions that the processed events must fulfill to trigger the correlation rule. Available settings vary based on the selected rule type.
- **Actions**—used to set the triggers that will activate when the conditions configured in the **Selectors** settings block are fulfilled. The Correlation rule resource must have at least one trigger. Available settings vary based on the selected rule type.
- **Correlators**—used for linking correlators. Available only for created correlation rules that are open for editing.

General tab

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—the tenant that owns the correlation rule.
- **Type** (required)—a drop-down list for selecting the type of correlation rule. Select **standard** if you want to create a standard correlation rule.
- **Identical fields** (required)—the event fields that should be grouped in a Bucket. The hash of the values of the selected fields is used as the Bucket key. If the selector (see below) triggers, the selected fields will be copied to the correlation event.

  If different selectors of the correlation rule use fields that have different values in events, do not specify these fields in the **Identical fields** section.

- **Unique fields**—event fields that should be sent to the Bucket. If this parameter is set, the Bucket will receive only unique events. The hash of the selected fields' values is used as the Bucket key.

  > You can use local variables in the **Identical fields** and **Unique fields** sections. To access a variable, its name must be preceded with the "$" character.
  > For an example of using local variables in these sections, refer to the rule provided with KUMA: R403_Access to malicious resources from a host with disabled protection or an out-of-date anti-virus database.

- **Rate limit**—maximum number of times a correlation rule can be triggered per second. The default value is 100.

  > If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the specific method used to count rule triggers in KUMA. In this case, try to increase the value of **Rate limit** to `1000000`, for example.

- **Window, sec** (required)—bucket lifetime, in seconds. Default value: 86,400 seconds (24 hours). This timer starts when the Bucket is created (when it receives the first event). The lifetime is not updated, and when it runs out, the **On timeout** trigger from the **Actions** group of settings is activated and the bucket is deleted. The **On every threshold** and **On subsequent thresholds** triggers can be activated more than once during the lifetime of the bucket.

- **Base events keep policy**—this drop-down list is used to specify which base events must be stored in the correlation event:

  - **first** (default value)—this option is used to store the first base event of the event collection that triggered creation of the correlation event.

  - **last**—this option is used to store the last base event of the event collection that triggered creation of the correlation event.

  - **all**—this option is used to store all base events of the event collection that triggered creation of the correlation event.

- **Priority**—base coefficient used to determine the importance of a correlation rule. The default value is **Low**.

- **Order by**—in this drop-down list, you can select the event field that will be used by the correlation rule selectors to track situational changes. This could be useful if you want to configure a correlation rule to be triggered when several types of events occur sequentially, for example.

- **Description**—the description of a resource. Up to 4,000 Unicode characters.

Selectors tab

A rule of the **standard** kind can have multiple selectors. You can add selectors by clicking the **Add selector** button and can remove them by clicking the **Delete selector** button. Selectors can be moved by clicking the ⠿ button.

For each selector, the following two tabs are available: **Settings** and **Local variables**.

The **Settings** tab contains the following settings:

- **Alias** (required)—unique name of the event group that meets the conditions of the selector. Must contain 1 to 128 Unicode characters.

- **Selector threshold (event count)** (required)—the number of events that must be received by the selector to trigger. The default value is 1.

- **Filter** (required)—used to set the criteria for determining events that should trigger the selector. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

  To create a filter:

  1. In the **Filter** drop-down list, select **Create new**.

  2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

  3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

  4. In the **Conditions** settings block, specify the conditions that the events must meet:

     a. Click the **Add condition** button.

     b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

     c. In the **operator** drop-down list, select an operator.

     Filter operators

     - **=**—the left operand equals the right operand.
     - **<**—the left operand is less than the right operand.
     - **<=**—the left operand is less than or equal to the right operand.
     - **>**—the left operand is greater than the right operand.
     - **>=**—the left operand is greater than or equal to the right operand.
     - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
     - **contains**—the left operand contains values of the right operand.
     - **startsWith**—the left operand starts with one of the values of the right operand.
     - **endsWith**—the left operand ends with one of the values of the right operand.
     - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.

- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Filtering based on data from the Extra event field

Conditions for filters based on data from the `Extra` event field:

- Condition—**If**.
- Left operand—**event field**.
- In this event field, you can specify one of the following values:
  - `Extra` field.
  - Value from the `Extra` field in the following format:

    `Extra.<field name>`

    For example, `Extra.app`.

    You must specify the value manually.
  - Value from the array written to the `Extra` field in the following format:

    `Extra.<field name>.<array element>`

    For example, `Extra.array.0`.

    The values in the array are numbered starting from 0. You must specify the value manually. To work with a value in the `Extra` field at a depth of 3 and lower, you must use backticks ``, for example, `` `Extra.lev1.lev2.lev3` ``.
- Operator – =.
- Right operand—**constant**.
- Value—the value by which you need to filter events.

The order of conditions specified in the selector filter of the correlation rule is significant and affects system performance. We recommend putting the most unique condition in the first place in the selector filter.

Consider two examples of selector filters that select successful authentication events in Microsoft Windows.

Selector filter 1:

Condition 1. DeviceProduct = Microsoft Windows

Condition 2. DeviceEventClassID = 4624

Selector filter 2:

Condition 1. DeviceEventClassID = 4624

Condition 2. DeviceProduct = Microsoft Windows

The order of conditions in Selector filter 2 is preferable because it causes less load on the system.

- **Recovery**—this check box must be selected when the Correlation rule must NOT trigger if a certain number of events are received from the selector. By default, this check box is cleared.

Select the **Local variables** tab and click **Add variable** to declare variables that you want to use within the limits of this correlation rule.

In the selector of the correlation rule, you can use regular expressions conforming to the RE2 standard.

Using regular expressions in correlation rules is computationally intensive compared to other operations. Therefore, when designing correlation rules, we recommend limiting the use of regular expressions to the necessary minimum and using other available operations.

To use a regular expression, you must use the `match` comparison operator. The regular expression must be placed in a constant. The use of capture groups in regular expressions is optional. For the correlation rule to trigger, the field text matched against the regexp must exactly match the regular expression.

For a primer on syntax and examples of correlation rules that use regular expressions in their selectors, see the following rules that are provided with KUMA:

- R105_04_Suspicious PowerShell commands. Suspected obfuscation.
- R333_Suspicious creation of files in the autorun folder.

Actions tab

A rule of the **standard** kind can have multiple triggers.

- **On first threshold**—this trigger activates when the Bucket registers the first triggering of the selector during the lifetime of the Bucket.
- **On subsequent thresholds**—this trigger activates when the Bucket registers the second and all subsequent triggering of the selector during the lifetime of the Bucket.
- **On every threshold**—this trigger activates every time the Bucket registers the triggering of the selector.
- **On timeout**—this trigger activates when the lifetime of the Bucket ends, and is linked to the selector with the **Recovery** check box selected. In other words, this trigger activates if the situation detected by the correlation rule is not resolved within the defined amount of time.

Every trigger is represented as a group of settings with the following parameters available:

- **Output**—if this check box is selected, the correlation event is sent for post-processing: for external enrichment outside the correlation rule, for response, and to destinations.
- **Loop to correlator**—if this check box is selected, the created correlation event is processed by the rule chain of the current correlator. This allows hierarchical correlation.

  If the **Output** and **Loop to correlator** check boxes are set, the correlation rule is sent to post-processing first, and then to the selectors of the current correlation rule.
- **Do not create alert**—if this check box is selected, an alert is not created when this correlation rule is triggered. If you do not want to create an alert when a correlation rule is triggered, but you still want to send a correlation event to the storage, select the **Output** and **No alert** check boxes. If you select only the **No alert** check box, a correlation event is not saved in the storage.
- Under **Enrichment**, you can modify the fields of correlation events by using enrichment rules. These enrichment rules are stored in the correlation rule where they were created. You can create multiple enrichment rules. Enrichment rules can be added or deleted by clicking the **Add enrichment** or **Remove enrichment** buttons, respectively.
  - **Source kind**—you can select the type of enrichment in this drop-down list. Depending on the selected type, you may see advanced settings that will also need to be completed.

Available types of enrichment:

- **constant**

  This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

  If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

  If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

- **dictionary**

  This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

  If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

  Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

  If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the " | " character when the dictionary is queried.

  Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **table**

  This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, click the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by clicking the **Add new element** button. Columns can be deleted by clicking the ✕ button.

- **event**

This type of enrichment is used when you need to write a value from another event field to the current event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Target field	The KUMA event field that you want to populate with the data.
Source field	The event field whose value is written to the target field.

Clicking 🔧 opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the KUMA event fields. You can reorder and delete created rules. To change the position of a rule, click ⠿ next to it. To delete a rule, click ✕ next to it.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.

- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
  - **Replace chars** specifies the sequence of characters to be replaced.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.
- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:
  - **Expression** is the RE2 regular expression whose results you want to replace.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

When using enrichment of events that have **event** selected as the **Source kind** and the extended event schema fields are used as arguments, the following special considerations apply:

- If the source extended event schema field has the `Array of strings` type, and the target extended event schema field has the `String` type, the values are written to the target extended event schema field in TSV format.

  Example: The `SA.StringArray` extended event schema field contains values: `"string1"`, `"string2"`, `"string3"`. An event enrichment operation is performed. The result of the event enrichment operation is written to the `DeviceCustomString1` extended event schema field. The `DeviceCustomString1` extended event schema field contains values: `["string1", "string2", "string3"]`.

- If the source and target extended event schema fields have the `Array of strings` type, values of the source extended event schema field are added to the values of the target extended event schema field, and the `","` character is used as the delimiter character.

  Example: The `SA.StringArrayOne` field of the extended event scheme contains the `["string1", "string2", "string3"]` values, and the `SA.StringArrayTwo` field of the extended event scheme contains the `["string4", "string5", "string6"]` values. An event enrichment operation is performed. The result of the event enrichment operation is written to the `SA.StringArrayTwo` field of the extended event scheme. The `SA.StringArrayTwo` extended event schema field contains values: `["string4", "string5", "string6", "string1", "string2", "string3"]`.

- **template**

  This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

- `{{.SA.StringArrayOne}}`
- `{{- range $index, $element := . SA.StringArrayOne -}}`
  `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

```
template {{toString .SA.StringArray}}
```

- **Debug**—you can use this toggle switch to enable logging of service operations.
- **Description**—the description of a resource. Up to 4,000 Unicode characters.

- **Categorization** settings group—used to change the categories of assets indicated in events. There can be several categorization rules. You can add or delete them by clicking the **Add categorization** or **Remove categorization** buttons. Only reactive categories can be added to assets or removed from assets.
  - **Operation**—this drop-down list is used to select the operation to perform on the category:
    - **Add**—assign the category to the asset.
    - **Delete**—unbind the asset from the category.
  - **Event field**—event field that indicates the asset requiring the operation.
  - **Category ID**—the drop-down list displays a tree of categories, in which you can select a category to perform the operation on. Clicking the row expands the list.
- **Active lists update** group of settings—used to assign the trigger for one or more operations with active lists. You can click the **Add active list action** and **Delete active list action** buttons to add or delete operations with active lists, respectively.

  Available settings:
  - **Name** (required)—this drop-down list is used to select the Active list resources.
  - **Operation** (required)—this drop-down list is used to select the operation that must be performed:
    - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the active list.
    - **Get**—get the Active list entry and write the values of the selected fields into the correlation event.
    - **Set**—write the values of the selected fields of the correlation event into the Active list by creating a new or updating an existing Active list entry. When the Active list entry is updated, the data is merged and only the specified fields are overwritten.
    - **Delete**—delete the Active list entry.

- **Key fields** (required)—this is the list of event fields used to create the Active list entry. It is also used as the Active list entry key.

  > The active list entry key depends on the available fields and does not depend on the order in which they are displayed in the KUMA Console.

- **Mapping** (required for **Get** and **Set** operations)—used to map Active list fields with events fields. More than one mapping rule can be set.

  - The left field is used to specify the Active list field.

    > The field must not contain special characters or numbers only.

  - The middle drop-down list is used to select event fields.
  - The right field can be used to assign a constant to the Active list field is the **Set** operation was selected.
  - Under Context table update, you can assign the trigger for one or more operations with context tables. You can click **Add context table action** or **Delete context table action** to add or delete operations with context tables.
  - Available settings:
  - Name (required)—this drop-down list is used to select context table resources.
  - Operation (required)—this drop-down list is used to select the operation that must be performed.
  - Sum—add a constant, the value of a correlation event field, or the value of a local variable to the value of the context table. This operation is used only for fields of number and float types.
  - Set—write the values of the selected fields of the correlation event into the context table by creating a new or updating an existing context table entry. When the context table entry is updated, the data is merged and only the specified fields are overwritten.
  - Get—get the fields of the context table and write the values of the specified fields into the correlation event. Table fields of the boolean type and lists of boolean values are excluded from mapping because the event does not contain boolean fields.
  - Merge—append the value of a correlation event field, local variable, or constant to the current value of a field of the context table.
  - Delete—delete the context table entry.
  - Key fields (required)—this is the list of event fields used to create the context table entry. It is also used as the key of the context table entry. As a key field, you can specify an event field or a local variable declared on the **Selectors** tab.
  - The composite key of the context table entry depends only on the values of fields and does not depend on the order in which they are displayed in the KUMA Console.
  - Mapping (required for all operations except **Delete**)—used to map context table fields to event fields or variables. More than one mapping rule can be set. You can specify the same context table field multiple times.

- The left field is used to specify the context table field.
- The field must not contain a field name that is already used in the mapping, tab characters, special characters, or only numerals. The maximum number of characters is 128. The name cannot begin with an underscore.
- The middle drop-down list is used to select event fields or a local variable.
- The right field can be used to assign a constant to the context table field is the **Set** operation was selected. **Merge** or **Sum**. The maximum number of characters is 1,024.

Correlators tab

- **Add**—Used when editing the created correlation rule. You can click **Add** to open the **Correlators** window and select a correlator from the list. After you click **OK**, the rule is linked to the selected correlator. You can select multiple correlators at the same time. The rule is added to the end of the execution queue. If you want to move the rule up in the execution queue, go to **Resources** - **Correlator** - <selected correlator> - **Edit correlator** - **Correlation**, select the check box next to the relevant rule and click the **Move up** or **Move down** buttons to reorder the rules as necessary.
- **Delete**—Used to unlink the correlation rule from the correlator.

Simple correlation rules

Simple correlation rules are used to define simple sequences of events.

The correlation rule window contains the following settings tabs:

- **General**—used to specify the main settings of the correlation rule. On this tab, you can select the type of correlation rule.
- **Selectors**—used to define the conditions that the processed events must fulfill to trigger the correlation rule. Available settings vary based on the selected rule kind.
- **Actions**—used to set the triggers that will activate when the conditions configured in the **Selectors** settings block are fulfilled. A correlation rule must have at least one trigger. Available settings vary based on the selected rule type.
- **Correlators**—used for linking correlators. Available only for created correlation rules that are open for editing.

General tab

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—the tenant that owns the correlation rule.
- **Type** (required)—a drop-down list for selecting the type of correlation rule. Select **simple** if you want to create a simple correlation rule.
- **Propagated fields** (required)—event fields used for event selection. If the selector (see below) is triggered, these fields will be written to the correlation event.

- **Rate limit**—maximum number of times a correlation rule can be triggered per second. The default value is 100.

  If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the specific method used to count rule triggers in KUMA. In this case, try to increase the value of **Rate limit** to `1000000`, for example.

- **Priority**—base coefficient used to determine the importance of a correlation rule. The default value is `Low`.

- **Description**—the description of a resource. Up to 4,000 Unicode characters.

Selectors tab

A rule of the **simple** kind can have only one selector for which the **Settings** and **Local variables** tabs are available.

The **Settings** tab contains settings with the **Filter** group of settings:

- **Filter** (required)—used to set the criteria for determining events that should trigger the selector. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

  To create a filter:

  1. In the **Filter** drop-down list, select **Create new**.
  2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.
  3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.
  4. In the **Conditions** settings block, specify the conditions that the events must meet:
     a. Click the **Add condition** button.
     b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.
     c. In the **operator** drop-down list, select an operator.

        Filter operators

        - **=**—the left operand equals the right operand.
        - **<**—the left operand is less than the right operand.
        - **<=**—the left operand is less than or equal to the right operand.
        - **>**—the left operand is greater than the right operand.
        - **>=**—the left operand is greater than or equal to the right operand.

- **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
- **contains**—the left operand contains values of the right operand.
- **startsWith**—the left operand starts with one of the values of the right operand.
- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Filtering based on data from the Extra event field

Conditions for filters based on data from the `Extra` event field:

- Condition—**If**.
- Left operand—**event field**.
- In this event field, you can specify one of the following values:
  - `Extra` field.
  - Value from the `Extra` field in the following format:

    `Extra.<field name>`

    For example, `Extra.app`.

    You must specify the value manually.
  - Value from the array written to the `Extra` field in the following format:

    `Extra.<field name>.<array element>`

    For example, `Extra.array.0`.

    The values in the array are numbered starting from 0. You must specify the value manually. To work with a value in the `Extra` field at a depth of 3 and lower, you must use backticks ``, for example, `` `Extra.lev1.lev2.lev3` ``.
- Operator – =.
- Right operand—**constant**.
- Value—the value by which you need to filter events.

The order of conditions specified in the selector filter of the correlation rule is significant and affects system performance. We recommend putting the most unique condition in the first place in the selector filter.

Consider two examples of selector filters that select successful authentication events in Microsoft Windows.

Selector filter 1:

Condition 1. DeviceProduct = Microsoft Windows

Condition 2. DeviceEventClassID = 4624

Selector filter 2:

Condition 1. DeviceEventClassID = 4624

Condition 2. DeviceProduct = Microsoft Windows

The order of conditions in Selector filter 2 is preferable because it causes less load on the system.

Select the **Local variables** tab and click **Add variable** to declare variables that you want to use within the limits of this correlation rule.

Actions tab

A rule of the **simple** kind can have only one trigger: **On every event**. It is activated every time the selector triggers.

Available parameters of the trigger:

- **Output**—if this check box is selected, the correlation event is sent for post-processing: for enrichment, for a response, and to destinations.

- **Loop to correlator**—if this check box is selected, the created correlation event is processed by the rule chain of the current correlator. This allows hierarchical correlation.

  If the **Output** and **Loop to correlator** check boxes are set, the correlation rule is sent to post-processing first, and then to the selectors of the current correlation rule.

- **Do not create alert**—if this check box is selected, an alert is not created when this correlation rule is triggered. If you do not want to create an alert when a correlation rule is triggered, but you still want to send a correlation event to the storage, select the **Output** and **No alert** check boxes. If you select only the **No alert** check box, a correlation event is not saved in the storage.

- **Enrichment** settings group—you can modify the fields of correlation events by using enrichment rules. These enrichment rules are stored in the correlation rule where they were created. You can create multiple enrichment rules. Enrichment rules can be added or deleted by clicking the **Add enrichment** or **Remove enrichment** buttons, respectively.

  - **Source kind**—you can select the type of enrichment in this drop-down list. Depending on the selected type, you may see advanced settings that will also need to be completed.

    Available types of enrichment:

    - **constant**

      This type of enrichment is used when a constant needs to be added to an event field. Available enrichment type settings are listed in the table below.

      Available enrichment type settings

Setting	Description
Constant	The value to be added to the event field. Maximum length of the value: 255 Unicode characters. If you leave this field blank, the existing event field value is removed.
Target field	The KUMA event field that you want to populate with the data.

      If you are using the event enrichment functions for extended schema fields of `String`, `Number`, or `Float` type with a constant, the constant is added to the field.

      If you are using the event enrichment functions for extended schema fields of `Array of strings`, `Array of numbers`, or `Array of floats` type with a constant, the constant is added to the elements of the array.

    - **dictionary**

      This type of enrichment is used if you need to add a value from the dictionary of the **Dictionary** type. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Dictionary name	The dictionary from which the values are to be taken.
Key fields	Event fields whose values are to be used for selecting a dictionary entry. To add an event field, click **Add field**. You can add multiple event fields.

If you are using event enrichment with the **dictionary** type selected as the **Source kind** setting, and an array field is specified in the **Key enrichment fields** setting, when an array is passed as the dictionary key, the array is serialized into a string in accordance with the rules of serializing a single value in the TSV format.

Example: The **Key fields** setting of the **enrichment** uses the `SA.StringArrayOne` extended schema field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`. The following values are passed to the dictionary as the key: `['a','b','c']`.

If the **Key enrichment fields** setting uses an array extended schema field and a regular event schema field, the field values are separated by the "|" character when the dictionary is queried.

Example: The **Key enrichment fields** setting uses the `SA.StringArrayOne` extended schema field and the `Code` string field. The `SA.StringArrayOne` extended schema field contains the values `"a"`, `"b"`, `"c"`, and the `Code` string field contains the `myCode` sequence of characters. The following values are passed to the dictionary as the key: `['a','b','c']|myCode`.

- **table**

This type of enrichment is used if you need to add a value from the dictionary of the **Table** type.

When this enrichment type is selected in the **Dictionary name** drop-down list, select the dictionary for providing the values. In the **Key fields** group of settings, click the **Add field** button to select the event fields whose values are used for dictionary entry selection.

In the **Mapping** table, configure the dictionary fields to provide data and the event fields to receive data:

- In the **Dictionary field** column, select the dictionary field. The available fields depend on the selected dictionary resource.
- In the **KUMA field** column, select the event field to which the value is written. For some of the selected fields (`*custom*` and `*flex*`), in the **Label** column, you can specify a name for the data written to them.

New table rows can be added by clicking the **Add new element** button. Columns can be deleted by clicking the ✕ button.

- **event**

This type of enrichment is used when you need to write a value from another event field to the current event field. Available enrichment type settings are listed in the table below.

Available enrichment type settings

Setting	Description
Target field	The KUMA event field that you want to populate with the data.
Source field	The event field whose value is written to the target field.

Clicking 🔧 opens the **Conversion** window, in which you can click **Add conversion** to create rules for modifying the source data before writing them to the KUMA event fields. You can reorder and delete created rules. To change the position of a rule, click ⠿ next to it. To delete a rule, click ✕ next to it.

Available conversions

Conversions are modifications that are applied to a value before it is written to the event field. You can select one of the following conversion types from the drop-down list:

- **entropy** is used for converting the value of the source field using the information entropy calculation function and placing the conversion result in the target field of the float type. The result of the conversion is a number. Calculating the information entropy allows detecting DNS tunnels or compromised passwords, for example, when a user enters the password instead of the login and the password gets logged in plain text.
- **lower**—is used to make all characters of the value lowercase
- **upper**—is used to make all characters of the value uppercase
- **regexp** – used to convert a value using a specified RE2 regular expression. When you select this type of conversion, a field is displayed in which you must specify the RE2 regular expression.
- **substring** is used to extract characters in a specified range of positions. When you select this type of conversion, the **Start** and **End** fields are displayed, in which you must specify the range of positions.
- **replace**—is used to replace specified character sequence with the other character sequence. When you select this type of conversion, the following fields are displayed:
  - **Replace chars** specifies the sequence of characters to be replaced.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.
- **trim** removes the specified characters from the beginning and from the end of the event field value. When you select this type of conversion, the **Chars** field is displayed in which you must specify the characters. For example, if a **trim** conversion with the `Micromon` value is applied to `Microsoft-Windows-Sysmon`, the new value is `soft-Windows-Sys`.
- **append** appends the specified characters to the end of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **prepend** prepends the specified characters to the beginning of the event field value. When you select this type of conversion, the **Constant** field is displayed in which you must specify the characters.
- **replace with regexp** is used to replace RE2 regular expression results with the specified character sequence. When you select this type of conversion, the following fields are displayed:
  - **Expression** is the RE2 regular expression whose results you want to replace.
  - **With chars** is the character sequence to be used instead of the character sequence being replaced.

- Converting encoded strings to text:
  - **decodeHexString**—used to convert a HEX string to text.
  - **decodeBase64String**—used to convert a Base64 string to text.
  - **decodeBase64URLString**—used to convert a Base64url string to text.

  When converting a corrupted string or if conversion error occur, corrupted data may be written to the event field.

  During event enrichment, if the length of the encoded string exceeds the size of the field of the normalized event, the string is truncated and is not decoded.

  If the length of the decoded string exceeds the size of the event field into which the decoded value is to be written, the string is truncated to fit the size of the event field.

Conversions when using the extended event schema

Whether or not a conversion can be used depends on the type of extended event schema field being used:

- For an additional field of the `String` type, all types of conversions are available.
- For fields of the `Number` and `Float` types, the following types of conversions are available: **regexp**, **substring**, **replace**, **trim**, **append**, **prepend**, **replaceWithRegexp**, **decodeHexString**, **decodeBase64String**, and **decodeBase64URLString**.
- For fields of `Array of strings`, `Array of numbers`, and `Array of floats` types, the following types of conversions are available: **append** and **prepend**.

When using enrichment of events that have **event** selected as the **Source kind** and the extended event schema fields are used as arguments, the following special considerations apply:

- If the source extended event schema field has the `Array of strings` type, and the target extended event schema field has the `String` type, the values are written to the target extended event schema field in TSV format.

  Example: The `SA.StringArray` extended event schema field contains values: `"string1"`, `"string2"`, `"string3"`. An event enrichment operation is performed. The result of the event enrichment operation is written to the `DeviceCustomString1` extended event schema field. The `DeviceCustomString1` extended event schema field contains values: `["string1", "string2", "string3"]`.

- If the source and target extended event schema fields have the `Array of strings` type, values of the source extended event schema field are added to the values of the target extended event schema field, and the `","` character is used as the delimiter character.

  Example: The `SA.StringArrayOne` field of the extended event scheme contains the `["string1", "string2", "string3"]` values, and the `SA.StringArrayTwo` field of the extended event scheme contains the `["string4", "string5", "string6"]` values. An event enrichment operation is performed. The result of the event enrichment operation is written to the `SA.StringArrayTwo` field of the extended event scheme. The `SA.StringArrayTwo` extended event schema field contains values: `["string4", "string5", "string6", "string1", "string2", "string3"]`.

- **template**

  This type of enrichment is used when you need to write a value obtained by processing Go templates into the event field. We recommend matching the value and the size of the field. Available enrichment type settings are listed in the table below.

  Available enrichment type settings

Setting	Description
Template	The Go template. Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field from which the value must be passed to the script, for example, `{{.DestinationAddress}} attacked from {{.SourceAddress}}`.
Target field	The KUMA event field that you want to populate with the data.

  If you are using enrichment of events that have **template** selected as the **Source kind**, and in which the target field has the `String` type, and the source field is an extended event schema field containing an array of strings, you can use one of the following examples for the template:

  - `{{.SA.StringArrayOne}}`
  - `{{- range $index, $element := . SA.StringArrayOne -}}`
    `{{- if $index}}, {{end}}"{{$element}}"{{- end -}}`

  To convert the data in an array field in a template into the TSV format, use the `toString` function, for example:

  ```
  template {{toString .SA.StringArray}}
  ```

- **Debug**—you can use this toggle switch to enable logging of service operations.
- **Description**—the description of a resource. Up to 4,000 Unicode characters.
- **Filter** settings block—lets you select which events will be forwarded for enrichment. Configuration is performed as described above.

- **Categorization** settings group—used to change the categories of assets indicated in events. There can be several categorization rules. You can add or delete them by clicking the **Add categorization** or **Remove categorization** buttons. Only reactive categories can be added to assets or removed from assets.
  - **Operation**—this drop-down list is used to select the operation to perform on the category:
    - **Add**—assign the category to the asset.
    - **Delete**—unbind the asset from the category.
  - **Event field**—event field that indicates the asset requiring the operation.
  - **Category ID**—the drop-down list displays a tree of categories, in which you can select a category to perform the operation on. Clicking the row expands the list.
- **Active lists update** group of settings—used to assign the trigger for one or more operations with active lists. You can click the **Add active list action** and **Delete active list action** buttons to add or delete operations with active lists, respectively.

  Available settings:
  - **Name** (required)—this drop-down list is used to select the active list.
  - **Operation** (required)—this drop-down list is used to select the operation that must be performed:
    - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the active list.
    - **Get**—get the Active list entry and write the values of the selected fields into the correlation event.
    - **Set**—write the values of the selected fields of the correlation event into the Active list by creating a new or updating an existing Active list entry. When the Active list entry is updated, the data is merged and only the specified fields are overwritten.
    - **Get**—get the Active list entry and write the values of the selected fields into the correlation event.
    - **Delete**—delete the Active list entry.
  - **Key fields** (required)—this is the list of event fields used to create the Active list entry. It is also used as the Active list entry key.

    > The active list entry key depends on the available fields and does not depend on the order in which they are displayed in the KUMA Console.

  - **Mapping** (required for **Get** and **Set** operations)—used to map Active list fields with events fields. More than one mapping rule can be set.
    - The left field is used to specify the Active list field.

      > The field must not contain special characters or numbers only.

    - The middle drop-down list is used to select event fields.
    - The right field can be used to assign a constant to the Active list field is the **Set** operation was selected.

- Under Context table update, you can assign the trigger for one or more operations with context tables. You can click **Add context table action** or **Delete context table action** to add or delete operations with context tables.

- Available settings:

- Name (required)—this drop-down list is used to select context table resources.

- Operation (required)—this drop-down list is used to select the operation that must be performed.

- Sum—add a constant, the value of a correlation event field, or the value of a local variable to the value of the context table. This operation is used only for fields of number and float types.

- Set—write the values of the selected fields of the correlation event into the context table by creating a new or updating an existing context table entry. When the context table entry is updated, the data is merged and only the specified fields are overwritten.

- Get—get the fields of the context table and write the values of the specified fields into the correlation event. Table fields of the boolean type and lists of boolean values are excluded from mapping because the event does not contain boolean fields.

- Merge—append the value of a correlation event field, local variable, or constant to the current value of a field of the context table.

- Delete—delete the context table entry.

- Key fields (required)—this is the list of event fields used to create the context table entry. It is also used as the key of the context table entry. As a key field, you can specify an event field or a local variable declared on the **Selectors** tab.

- The composite key of the context table entry depends only on the values of fields and does not depend on the order in which they are displayed in the KUMA Console.

- Mapping (required for all operations except **Delete**)—used to map context table fields to event fields or variables. More than one mapping rule can be set. You can specify the same context table field multiple times.

- The left field is used to specify the context table field.

- The field must not contain a field name that is already used in the mapping, tab characters, special characters, or only numerals. The maximum number of characters is 128. The name cannot begin with an underscore.

- The middle drop-down list is used to select event fields or a local variable.

- The right field can be used to assign a constant to the context table field is the **Set** operation was selected. **Merge** or **Sum**. The maximum number of characters is 1,024.

Correlators tab

- **Add**—Used when editing the created correlation rule. You can click **Add** to open the **Correlators** window and select a correlator from the list. After you click **OK**, the rule is linked to the selected correlator. You can select multiple correlators at the same time. The rule is added to the end of the execution queue. If you want to move the rule up in the execution queue, go to **Resources** - **Correlator** - <selected correlator> - **Edit correlator** - **Correlation**, select the check box next to the relevant rule and click the Move up or Move down buttons to reorder the rules as necessary.

- **Delete**—Used to unlink the correlation rule from the correlator.

Operational correlation rules

Operational correlation rules are used for working with active lists.

The correlation rule window contains the following tabs:

- **General**—used to specify the main settings of the correlation rule. On this tab, you can select the type of correlation rule.
- **Selectors**—used to define the conditions that the processed events must fulfill to trigger the correlation rule. Available settings vary based on the selected rule type.
- **Actions**—used to set the triggers that will activate when the conditions configured in the **Selectors** settings block are fulfilled. A correlation rule must have at least one trigger. Available settings vary based on the selected rule type.
- **Correlators**—used for linking correlators. Available only for created correlation rules that are open for editing.

General tab

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—the tenant that owns the correlation rule.
- **Type** (required)—a drop-down list for selecting the type of correlation rule. Select **operational** if you want to create an operational correlation rule.
- **Rate limit**—maximum number of times a correlation rule can be triggered per second. The default value is 100.

  If correlation rules employing complex logic for pattern detection are not triggered, this may be due to the specific method used to count rule triggers in KUMA. In this case, try to increase the value of **Rate limit** to `1000000`, for example.

- **Description**—the description of a resource. Up to 4,000 Unicode characters.

Selectors tab

A rule of the **operational** kind can have only one selector for which the **Settings** and **Local variables** tabs are available.

The **Settings** tab contains settings with the **Filter** group of settings:

- **Filter** (required)—used to set the criteria for determining events that should trigger the selector. You can select an existing filter from the drop-down list or **create** a new filter.

  Creating a filter in resources

To create a filter:

1. In the **Filter** drop-down list, select **Create new**.

2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default.

3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters.

4. In the **Conditions** settings block, specify the conditions that the events must meet:

   a. Click the **Add condition** button.

   b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field.

   c. In the **operator** drop-down list, select an operator.

   Filter operators

   - **=**—the left operand equals the right operand.
   - **<**—the left operand is less than the right operand.
   - **<=**—the left operand is less than or equal to the right operand.
   - **>**—the left operand is greater than the right operand.
   - **>=**—the left operand is greater than or equal to the right operand.
   - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
   - **contains**—the left operand contains values of the right operand.
   - **startsWith**—the left operand starts with one of the values of the right operand.
   - **endsWith**—the left operand ends with one of the values of the right operand.
   - **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
   - **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

     The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

     If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.

- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.

- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.

- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.

- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.

- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.

- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.

- **inContextTable**—presence of the entry in the specified context table.

- **intersect**—presence in the left operand of the list items specified in the right operand.

d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default.

e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

You can add multiple conditions or a group of conditions.

5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button.

6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Filtering based on data from the Extra event field

Conditions for filters based on data from the `Extra` event field:

- Condition—**If**.

- Left operand—**event field**.

- In this event field, you can specify one of the following values:

  - `Extra` field.

  - Value from the `Extra` field in the following format:

`Extra.<field name>`

For example, `Extra.app`.

You must specify the value manually.

- Value from the array written to the `Extra` field in the following format:

  `Extra.<field name>.<array element>`

  For example, `Extra.array.0`.

  The values in the array are numbered starting from 0. You must specify the value manually. To work with a value in the `Extra` field at a depth of 3 and lower, you must use backticks ``, for example, `` `Extra.lev1.lev2.lev3` ``.

- Operator – =.
- Right operand—**constant**.
- Value—the value by which you need to filter events.

On the **Local variables** tab, click **Add variable** to declare variables that you want to use within the limits of this correlation rule.

Actions tab

A rule of the **operational** kind can have only one trigger: **On every event**. It is activated every time the selector triggers.

Available parameters of the trigger:

- **Active lists update** group of settings—used to assign the trigger for one or more operations with active lists. You can click the **Add active list action** and **Delete active list action** buttons to add or delete operations with active lists, respectively.

  Available settings:

  - **Name** (required)—this drop-down list is used to select the active list.
  - **Operation** (required)—this drop-down list is used to select the operation that must be performed:
    - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the active list.
    - **Set**—write the values of the selected fields of the correlation event into the Active list by creating a new or updating an existing Active list entry. When the Active list entry is updated, the data is merged and only the specified fields are overwritten.
    - **Delete**—delete the Active list entry.
  - **Key fields** (required)—this is the list of event fields used to create the Active list entry. It is also used as the Active list entry key.

    The active list entry key depends on the available fields and does not depend on the order in which they are displayed in the KUMA Console.

- **Mapping** (required for **Set** operation)—used to map active list fields with event fields. More than one mapping rule can be set.
  - The left field is used to specify the Active list field.

    > The field must not contain special characters or numbers only.

  - The middle drop-down list is used to select event fields.
  - The right field can be used to assign a constant to the Active list field is the **Set** operation was selected.

- Under **Context table update**, you can assign the trigger for one or more operations with context tables. You can click **Add context table action** or **Delete context table action** to add or delete operations with context tables, respectively.

  Available settings:

  - **Name** (required)—this drop-down list is used to select context table resources.
  - **Operation** (required)—this drop-down list is used to select the operation that must be performed.
    - **Sum**—add a constant, the value of a correlation event field, or the value of a local variable to the value of the context table. This operation is used only for fields of number and float types.
    - **Set**—write the values of the selected fields of the correlation event into the context table by creating a new or updating an existing context table entry. When the context table entry is updated, the data is merged and only the specified fields are overwritten.
    - **Merge**—append the value of a correlation event field, local variable, or constant to the current value of a field of the context table.
    - **Delete**—delete the context table entry.
  - **Key fields** (required)—this is the list of event fields used to create the context table entry. It is also used as the key of the context table entry. As a key field, you can specify an event field or a local variable declared on the **Selectors** tab.

    > The composite key of the context table entry depends only on the values of fields and does not depend on the order in which they are displayed in the KUMA Console.

  - **Mapping** (required for all operations except **Delete**)—used to map context table fields to event fields or variables. More than one mapping rule can be set. You can specify the same context table field multiple times.
    - The left field is used to specify the context table field.

      The field must not contain a field name that is already used in the mapping, tab characters, special characters, or only numerals. The maximum number of characters is 128. The name cannot begin with an underscore.

    - The middle drop-down list is used to select event fields or a local variable.
    - You can use the right field to assign a constant to the context table field. The maximum number of characters is 1024.

Correlators tab

- **Add**—Used when editing the created correlation rule. You can click **Add** to open the **Correlators** window and select a correlator from the list. After you click **OK**, the rule is linked to the selected correlator. You can select multiple correlators at the same time. The rule is added to the end of the execution queue. If you want to move the rule up in the execution queue, go to **Resources** - **Correlator** - <selected correlator> - **Edit correlator** - **Correlation**, select the check box next to the relevant rule and click the Move up or Move down buttons to reorder the rules as necessary.

- **Delete**—Used to unlink the correlation rule from the correlator.

Correlation rules of the 'periodical' type

Correlation rules of the **periodical** type are used to detect unauthorized use of accounts by detecting anomalous user activity that may indicate a compromised account. This rule can be created only when using the correlator-ng correlator.

The mechanism for analyzing user authentication events creates a profile of the users' normal behavior and identifies deviations from this behavior based on statistical indicators. Anomaly detection is performed at regular intervals based on the accumulated data, and not at the moment when each event is received.

A **periodical** correlation rule works in two stages:

1. **Accumulation and aggregation of data**

   At the first stage, the correlation rule accumulates user behavior statistics. This stage corresponds to the cold start period, during which no anomalies are detected. The accumulation of statistics begins with the first event that is passed to the rule. None of the events created earlier are taken into account.

   At the data accumulation and aggregation stage, the following takes place:

   - Events for successful and unsuccessful attempts to log in to the user account are received.
   - The addresses from which and to which the logon is performed are extracted.
   - The data is aggregated on a per-user basis: unique values are saved, new addresses are counted for each authentication events grouping period.

   The duration of the cold start period is fixed and is specified in the configuration of the correlation rule. A cold start can be performed in the following ways:

   - By processing the event stream arriving at the correlator.
   - By using the retroscan procedure to download events from storage.
   - By using a combined method, in which some data is downloaded via retroscan, and the remaining data is accumulated by processing the event stream.

   When using the retroscan procedure, the correlation rule processes downloaded events in the same way as events arriving from the stream. If the storage contains user authentication events for the minimum period required for statistics accumulation, the rule begins periodically detecting anomalies immediately after these events are downloaded.

   It is acceptable to load data for a period not longer than the cold start duration, and then continue accumulating statistics by processing the event stream until the cold start period is completed.

2. **Anomaly detection**

After the end of the cold start period, the rule starts periodically analyzing the accumulated data at certain intervals. For each user and each analyzed indicator:

- The value of the indicator for the last authentication event grouping period is used.

  If the anomaly detection period coincides with the grouping period, the value of the indicator for the most recently completed period is used for analysis. If the detection period is shorter than the grouping period, the analysis is performed based on data from the current (ongoing) period, which allows getting results before the whole grouping period runs out.

- Statistical indicators are calculated.

- The results are compared to the thresholds that are configured in the correlation rule.

If indicators of anomalous user activity exceed any thresholds, a correlation event is generated. If necessary, the rule can additionally create an alert.

Indicators of anomalous user activity

To detect anomalies, two indicators are used together:

- **Relative Score**

  This indicator shows by how many times the user's current activity exceeds the user's average activity level, and is calculated using the following formula:

  Relative Score = (**Current activity** + 1) / (**Average activity** + 1)

  Where:

  Current activity is the current activity value.

  Average activity is the average activity level during the period for which the indicator is calculated.

- **Z-Score (standard score)**

  This indicator quantifies the deviation of the current value from the average value, taking into account dispersion, and is calculated using the following formula:

  Z-Score = (**Current activity** – **Average activity**) / **Standard deviation**

  Where:

  Current activity is the current activity value.

  Average activity is the average activity level during the period for which the indicator is calculated.

  Standard deviation is a measure of dispersion of activity values during the choosing period.

Abnormal user activity is logged if the value of one of the indicators (**Relative Score** or **Z-Score**) exceeds the specified threshold. For each analyzed property, the rule selects the most appropriate indicator depending on the user behavior statistics.

Using two indicators allows evaluating anomalies in different behavior scenarios, reducing the number of false positives. When an anomaly is detected, the rule creates a correlation event with information about the user, time, analyzed property, indicator type, indicator value, the threshold that was exceeded, as well as anomalous values for anomaly analysis.

Creating a correlation rule in the correlator-ng correlator

You can create a correlation rule of the **periodical** type only using the correlator-ng correlator.

When using this correlator, keep in mind that it uses strict data typing, supports PCRE regular expressions, and automatically converts existing rules to the new format upon first launch. A data type error or incorrect expression can prevent the rule from running.

*To create a correlation rule of the **periodical** type:*

1. Create a correlator service in OSMP Console.

2. Run correlator-ng:
   ```
   /opt/kaspersky/kuma/kuma ng correlator --core <KUMA_Core_URL> --id
   <correlator_service_ID> --api.port <correlator_API_port> --wd
   <correlator_working_directory>
   ```
   To view the list of available options, run the following command:
   ```
   /opt/kaspersky/kuma/kuma ng correlator --help
   ```
   When the correlator is started for the first time, configuration files are automatically created in the working directory, including the 'config' directory.

3. Stop the correlator.

4. Go to the config directory and specify the settings of the correlation rule that you want to create in the correlation-rules.yaml file.

   Description of available settings

   Example configuration of a correlation rule of the periodical type

```
id: 69df6a1b-262b-48f0-aab4-be15e822fc86
name: Lateral movement detection test
kind: periodical
selectors:
  tag: rule fired
  filter: null
  sql:
  (
  DeviceEventClassID = '77777'
  AND DeviceProduct = 'Windows'
  )
  recovery: false
  threshold: 0
identicalFields:
  SourceUserName
window: 0
keepPolicy: all
orderField: Timestamp
actions:
  onEveryThreshold:
    output: true
    loop: false
    noAlert: true
    activeLists: null
    contextTables: null
priority: 1
lateralMovementDetection:
  thresholds:
    sourceAddress:
      zScore: 12
      relativeScore: 8.5
    destinationAddress:
      zScore: 25
      relativeScore: 45
  minSamples: 15
  detectionPeriod: 60
  eventGroupingPeriod: 1
  propertyValuesCollectingPeriod: 30
  countersCollectingPeriod: 30
  retentionPeriod: 60
  propertyFieldsMapping:
    sourceEventField: DeviceCustomString1
    destinationProperty: sourceAddress
    sourceEventField: DeviceCustomString2
    destinationProperty: destinationAddress
```

5. Save the changes to the file.

6. Run correlator-ng again:
   `/opt/kaspersky/kuma/kuma ng correlator --core <KUMA_Core_URL> --id <correlator_service_ID> --api.port <correlator_API_port> --wd <correlator_working_directory>`

Settings in the correlation-rules.yaml file

This table lists the settings that can be configured in a correlation rule of the **periodical** type in the correlation-rules.yaml file.

Description of settings in the correlation-rules.yaml configuration file

Setting	Description
`identicalFields`	This setting specifies one or more fields of the KUMA event schema that are used to merge events for the purpose of analyzing the activity of an individual user. Example value: `SourceUserName`
`zScore`	Standard score (Z-Score) threshold. This setting determines how much the current activity value of a user must deviate from the average activity value, taking into account the dispersion of the data. This setting is configured separately for each detection property: • For the `sourceAddress` detection property: the minimum value is `3`, the recommended value is `12`. • For the `destinationAddress` detection property: the minimum value is `3`, the recommended value is `25`.
`relativeScore`	Relative Score threshold. This setting determines by how many times the user's current activity must exceed the average activity level for a certain period. This setting is configured separately for each detection property: • For the `sourceAddress` detection property: the minimum value is `3`, the recommended value is `8.5`. • For the `destinationAddress` detection property: the minimum value is `3`, the recommended value is `45`.
`minSamples`	The minimum number of user authentication events required to analyze anomalous activity. This setting determines the number of authentication events that must be accumulated for the correlation rule to be able to analyze anomalous activity. The minimum value is `15`, the recommended value is `15`. The maximum value is `60`, provided that the maximum values are specified for `eventGroupingPeriod` and `countersCollectingPeriod`.
`detectionPeriod`	Detection period. This setting determines the time interval in minutes after which the rule analyzes the accumulated user authentication events to detect anomalous activity. The minimum value is `15`, the recommended value is `60`. The maximum value is `10080`, provided that the maximum value is specified for `eventGroupingPeriod`.
`eventGroupingPeriod`	Event grouping period. This setting determines the interval in days for which the user authentication events are aggregated to be analyzed for anomalous activity. The minimum value is `1`, the maximum value is `7`, and the recommended value is `1`.
`propertyValuesCollectingPeriod`	Period of historical values collection. This setting determines the interval in days during which the unique values of the analyzed user properties are accumulated to be analyzed for anomalous activity. The minimum value is `15`, the maximum value is `15`, and the recommended value is `30`.
`countersCollectingPeriod`	The counter collection period. This setting determines the interval in days for which data is accumulated to generate indicators of abnormal activity. The value can be calculated using the following formula: `countersCollectingPeriod` = number of periods * `eventGroupingPeriod`. The minimum value is `15`, the maximum value is `420`, and the recommended value is `30`.
`retentionPeriod`	Retention period for correlation rule data. This setting determines the interval in days for which the data accumulated by the rule is retained for subsequent analysis. The value can be calculated using the following formula: `retentionPeriod` = number of periods * `eventGroupingPeriod`. The minimum value is `30`, the maximum value is `420`, and the recommended value is `60`.
`propertyFieldsMapping`	Field mappings for the values of the analyzed properties. This setting determines the mapping between a KUMA event field and the type of the analyzed property to detect anomalous user activity.

Setting	Description
`sourceEventField`	KUMA event field from the event schema. Example value: `DeviceCustomString1`, `DeviceCustomString2`.
`destinationProperty`	Analyzed properties. Possible values: `sourceAddress`, `destinationAddress`.

Variables in correlators

If tracking values in event fields, active lists, or dictionaries is not enough to cover some specific security scenarios, you can use global and local *variables*. You can use them to take various actions on the values received by the correlators by implementing complex logic for threat detection. Variables can be declared in the correlator (*global variables*) or in the correlation rule (*local variables*) by assigning a function to them, then querying them from correlation rules as if they were ordinary event fields and receiving the triggered function result in response.

Usage scope of variables:

- When searching for identical or unique field values in correlation rules.
- In the correlation rule selectors, in the filters of the conditions under which the correlation rule must be triggered.
- When enriching correlation events. Select **Event** as the source type.
- When populating active lists with values.

Variables can be queried the same way as event fields by preceding their names with the $ character.

Local variables in identical and unique fields

You can use local variables in the **Identical fields** and **Unique fields** sections of 'standard' type correlation rules. To use a local variable, its name must be preceded with the "$" character.

For an example of using local variables in the **Identical fields** and **Unique fields** sections, refer to the rule provided with KUMA: R403_Access to malicious resources from a host with disabled protection or an out-of-date anti-virus database.

Local variables in selector

To use a local variable in a selector:

1. Add a local variable to the rule.
2. In the **Correlation rules** window, go to the **General** tab and add the created local variable to the **Identical fields** section. Prefix the local variable name with a "$" character.
3. In **Correlation rules** window, go to the **Selectors** tab, select an existing filter or create a new filter and click **Add condition**.
4. Select the **event field** as the operand.

5. Select the local variable as the event field value and prefix the variable name with a "$" character.

6. Specify the remaining filter settings.

7. Click **Save**.

For an example of using local variables, refer to the rule provided with KUMA: R403_Access to malicious resources from a host with disabled protection or an out-of-date anti-virus database.

Local Variables in event enrichment

You can use 'standard' and 'simple' correlation rules to enrich events with local variables.

Enrichment with text and numbers

You can enrich events with text (strings). To do so, you can use functions that modify strings: to_lower, to_upper, str_join, append, prepend, substring, tr, replace, str_join.

You can enrich events with numbers. To do so, you can use the following functions: addition ("+"), subtraction ("-"), multiplication ("*"), division ("/"), round, ceil, floor, abs, pow.

You can also use regular expressions to manage data in local variables.

Using regular expressions in correlation rules is computationally intensive compared to other operations. Therefore, when designing correlation rules, we recommend limiting the use of regular expressions to the necessary minimum and using other available operations.

Timestamp enrichment

You can enrich events with timestamps (date and time). To do so, you can use functions that let you get or modify timestamps: now, extract_from_timestamp, parse_timestamp, format_timestamp, truncate_timestamp, time_diff.

Operations with active lists and tables

You can enrich events with local variables and data from active lists and tables.

To enrich events with data from an active list, use the active_list, active_list_dyn functions.

To enrich events with data from a table, use the table_dict, dict functions.

You can create conditional statements by using the 'conditional' function in local variables. In this way, the variable can return one of the values depending on what data was received for processing.

Enriching events with a local variable

To use a local variable to enrich events:

1. Add a local variable to the rule.
2. In the **Correlation rules** window, go to the **General** tab and add the created local variable to the **Identical fields** section. Prefix the local variable name with a "$" character.
3. In the **Correlation rules** window, go to the **Actions** tab, and under **Enrichment**, in the **Source kind** drop-down list, select **Event**.
4. From the **Target field** drop-down list, select the KUMA event field to which you want to pass the value of the local variable.
5. From the **Source field** drop-down list, select a local variable. Prefix the local variable name with a "$" character.
6. Specify the remaining rule settings.
7. Click **Save**.

Local variables in active list enrichment

You can use local variables to enrich active lists.

To enrich the active list with a local variable:

1. Add a local variable to the rule.
2. In the **Correlation rules** window, go to the **General** tab and add the created local variable to the **Identical fields** section. Prefix the local variable name with a "$" character.
3. In the **Correlation rules** window, go to the **Actions** tab and under **Active lists update**, add the local variable to the **Key fields** field. Prefix the local variable name with a "$" character.
4. Under **Mapping**, specify the correspondence between the event fields and the active list fields.
5. Click the **Save** button.

Properties of variables

Local and global variables

The properties of global variables differ from the properties of local variables.

Global variables:

- Global variables are declared at the correlator level and are applied only within the scope of this correlator.
- The global variables of the correlator can be queried from all correlation rules that are specified in it.

- In standard correlation rules, the same global variable can take different values in each selector.
- It is not possible to transfer global variables between different correlators.

Local variables:

- Local variables are declared at the correlation rule level and are applied only within the limits of this rule.
- In standard correlation rules, the scope of a local variable consists of only the selector in which the variable was declared.
- Local variables can be declared in any type of correlation rule.
- Local variables cannot be transferred between rules or selectors.
- A local variable cannot be used as a global variable.

Variables used in various types of correlation rules

- In **operational** correlation rules, on the **Actions** tab, you can specify all variables available or declared in this rule.
- In **standard** correlation rules, on the **Actions** tab, you can provide only those variables specified in these rules on the **General** tab, in the **Identical fields** field.
- In **simple** correlation rules, on the **Actions** tab, you can provide only those variables specified in these rules on the **General** tab, in the **Inherited Fields** field.

Requirements for variables

When adding a variable function, you must first specify the name of the function, and then list its parameters in parentheses. Basic mathematical operations (addition, subtraction, multiplication, division) are an exception to this requirement. When these operations are used, parentheses are used to designate the severity of the operations.

Requirements for function names:

- Must be unique within the correlator.
- Must contain 1 to 128 Unicode characters.
- Must not begin with the character $.
- Must be written in camelCase or CamelCase.

Special considerations when specifying functions of variables:

- The sequence of parameters is important.
- Parameters are separated by a comma: `,`.
- String parameters are passed in single quotes: `'`.
- Event field names and variables are specified without quotation marks.

- When querying a variable as a parameter, add the `$` character before its name.
- You do not need to add a space between parameters.
- In all functions in which a variable can be used as parameters, nested functions can be created.

Functions of variables

Operations with active lists and dictionaries

active_list and active_list_dyn functions

These functions allow you to receive information from an active list and dynamically generate a field name for an active list and key.

You must specify the parameters in the following sequence:

1. Name of the active list.
2. Expression that returns the field name of the active list.
3. One or more expressions whose results are used to generate the key.

Usage example	Result
`active_list('Test', to_lower('DeviceHostName'), to_lower(DeviceCustomString2), to_lower(DeviceCustomString1))`	Gets the field value of the active list.

Use these functions to query the active list of the shared tenant from a variable. To do so, add the @Shared suffix after the name of the active list (case sensitive). For example, active_list('exampleActiveList@Shared', 'score', SourceAddress, SourceUserName).

table_dict function

Gets information about the value in the specified column of a dictionary of the table type.

You must specify the parameters in the following sequence:

1. Dictionary name.
2. Dictionary column name.
3. One or more expressions whose results are used to generate the dictionary row key.

Usage example	Result
`table_dict('exampleTableDict', 'office', SourceUserName)`	Gets data from the `exampleTableDict` dictionary from the row with the `SourceUserName` key in the `office` column.
`table_dict('exampleTableDict', 'office', SourceAddress, to_lower(SourceUserName))`	Gets data from the `exampleTableDict` dictionary from a composite key string from the `SourceAddress` field value and the lowercase value of the `SourceUserName` field from the `office` column.

Use this function to access the dictionary of the shared tenant from a variable. To do so, add the `@Shared` suffix after the name of the active list (case sensitive). For example, `table_dict('exampleTableDict@Shared', 'office', SourceUserName)`.

dict function

Gets information about the value in the specified column of a dictionary of the dictionary type.

You must specify the parameters in the following sequence:

1. Dictionary name.
2. One or more expressions whose results are used to generate the dictionary row key.

Usage example	Result
`dict('exampleDictionary', SourceAddress)`	Gets data from `exampleDictionary` from the row with the `SourceAddress` key.
`dict('exampleDictionary', SourceAddress, to_lower(SourceUserName))`	Gets data from the `exampleDictionary` from a composite key string from the `SourceAddress` field value and the lowercase value of the `SourceUserName` field.

Use this function to access the dictionary of the shared tenant from a variable. To do so, add the `@Shared` suffix after the name of the active list (case sensitive). For example, `dict('exampleDictionary@Shared', SourceAddress)`.

Operations with context tables

context_table function

Returns the value of the specified field in the base type (for example, integer, array of integers).

You must specify the parameters in the following sequence:

1. Name of the context table. The name must be specified.
2. Expression that returns the field name of context table.
3. Expression that returns the name of key field 1 of the context table.
4. Expression that returns the value of key field 1 of the context table.

The function must contain at least 4 parameters.

Usage example	Result
`context_table('tbl1', 'list_field1', 'key1', 'key1_val')`	`Get the value of the specified field. If the context table or context table field does not exist, an empty string is returned.`

len function

Returns the length of a string or array.

The function returns the length of the array if the passed array is of one of the following types:

- array of integers
- array of floats

- array of strings
- array of booleans

If an array of a different type is passed, the data of the array is cast to the string type, and the function returns the length of the resulting string.

Usage examples
`len(context_table('tbl1', 'list_field1', 'key1', 'key1_val'))`
`len(DeviceCustomString1)`

distinct_items function

Returns a list of unique elements in an array.

The function returns the list of unique elements of the array if the passed array is of one of the following types:

- array of integers
- array of floats
- array of strings
- array of booleans

If an array of a different type is passed, the data of the array is cast to the string type, and the function returns a string consisting of the unique characters from the original string.

Usage examples
`distinct_items(context_table('tbl1', 'list_field1', 'key1', 'key1_val'))`
`distinct_items(DeviceCustomString1)`

sort_items function

Returns a sorted list of array elements.

You must specify the parameters in the following sequence:

1. Expression that returns the object of the sorting.
2. Sorting order Possible values: `asc`, `desc`. If the parameter is not specified, the default value is `asc`.

The function returns the list of sorted elements of the array if the passed array is of one of the following types:

- array of integers
- array of floats
- array of strings

For a boolean array, the function returns the list of array elements in the original order.

If an array of a different type is passed, the data of the array is cast to the string type, and the function returns a string of sorted characters.

Usage examples
`sort_items(context_table('tbl1', 'list_field1', 'key1', 'key1_val'), 'asc')`
`sort_items(DeviceCustomString1)`

item function

Returns the array element with the specified index or the character of a string with the specified index if an array of integers, floats, strings, or boolean values is passed.

You must specify the parameters in the following sequence:

1. Expression that returns the object of the indexing.
2. Expression that returns the index of the element or character.

The function must contain at least 2 parameters.

The function returns the array element with the specified index or the string character with the specified index if the index falls within the range of the array and the passed array is of one of the following types:

- array of integers
- array of floats
- array of strings
- array of booleans

If an array of a different type is passed and the index falls within the range of the array, the data is cast to the string type, and the function returns the string character with the specified index. If an array of a different type is passed and the index is outside the range of the array, the function returns an empty string.

Usage examples
`item(context_table('tbl1', 'list_field1', 'key1', 'key1_val'), 1)`
`item(DeviceCustomString1, 0)`

Operations with strings

to_lower function

Converts characters in a string to lowercase. Supported for standard fields and extended event schema fields of the `string` type.

A string can be passed as a string, field name or variable.

Usage examples
`to_lower(SourceUserName)`

`to_lower('SomeText')`
`to_lower($otherVariable)`

to_upper function

Converts characters in a string to uppercase. Supported for standard fields and extended event schema fields of the `string` type. A string can be passed as a string, field name or variable.

Usage examples
`to_upper(SourceUserName)`
`to_upper('SomeText')`
`to_upper($otherVariable)`

append function

Adds characters to the end of a string. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.
2. Added string.

Strings can be passed as a string, field name or variable.

Usage examples	Usage result
`append(Message, '123')`	The string `123` is added to the end of this string from the `Message` field.
`append($otherVariable, 'text')`	The string `text` is added to the end of this string from the variable `otherVariable`.
`append(Message, $otherVariable)`	A string from `otherVariable` is added to the end of this string from the `Message` field.

prepend function

Adds characters to the beginning of a string. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.
2. Added string.

Strings can be passed as a string, field name or variable.

Usage examples	Usage result
`prepend(Message, '123')`	The string `123` is added to the beginning of this string from the `Message` field.

`prepend($otherVariable, 'text')`	The string `text` is added to the beginning of this string from `otherVariable`.
`prepend(Message, $otherVariable)`	A string from `otherVariable` is added to the beginning of this string from the `Message` field.

substring function

Returns a substring from a string. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.
2. Substring start position (natural number or 0).
3. (Optional) substring end position.

Strings can be passed as a string, field name or variable. If the position number is greater than the original data string length, an empty string is returned.

Usage examples	Usage result
`substring(Message, 2)`	Returns a part of the string from the `Message` field: from 3 characters to the end.
`substring($otherVariable, 2, 5)`	Returns a part of the string from the `otherVariable` variable: from 3 to 6 characters.
`substring(Message, 0, len(Message) - 1)`	Returns the entire string from the `Message` field except the last character.

index_of function

The `index_of` function returns the byte offset of a character or substring in a string; the first character in the string has index 0. If the function does not find the substring, the function returns a negative value.

If the string has non-ASCII characters, the returned byte offset will not correspond to the number of characters preceding the substring you are searching for.

The function accepts the following parameters:

- As source data, an event field, another variable, or constant.
- Any expression out of those that are available in local variables.

To use this function, you must specify the parameters in the following order:

1. Character or substring whose position you want to find.
2. String to be searched.

Usage examples	Usage result
`index_of('@', SourceUserName)`	The function looks for the `@` character in the `SourceUserName` field. The `SourceUserName` field contains the `user@example.com` string. Result = 4 The function returns the index of the first occurrence of the character in the string. The first character in the string has index 0.

`index_of('m', SourceUserName)`	The function looks for the `m` character in the `SourceUserName` field. The `SourceUserName` field contains the `user@example.com` string. Result = 8 The function returns the index of the first occurrence of the character in the string. The first character in the string has index 0.

last_index_of function

The `last_index_of` function returns the position of the last occurrence of a character or substring in a string; the first character in the string has index 0. If the function does not find the substring, the function returns -9223372036854775808.

The function accepts the following parameters:

- As source data, an event field, another variable, or constant.
- Any expression out of those that are available in local variables.

To use this function, you must specify the parameters in the following order:

1. Character or substring whose position you want to find.
2. String to be searched.

Usage examples	Usage result
`last_index_of('m', SourceUserName)`	The function looks for the `m` character in the `SourceUserName` field. The `SourceUserName` field contains the `user@example.com` string. Result = 15 The function returns the index of the last occurrence of the character in the string. The first character in the string has index 0.

tr function

Removes the specified characters from the beginning and end of a string. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.
2. (Optional) string that should be removed from the beginning and end of the original string.

Strings can be passed as a string, field name or variable. If you do not specify a string to be deleted, spaces will be removed from the beginning and end of the original string.

Usage examples	Usage result
`tr(Message)`	Spaces have been removed from the beginning and end of the string from the `Message` field.
`tr($otherVariable, '_')`	If the `otherVariable` variable has the `_test_` value, the string `_test_` is returned.
`tr(Message, '@example.com')`	If the `Message` event field contains the string `user@example.com`, the string `user` is returned.

replace function

Replaces all occurrences of character sequence A in a string with character sequence B. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.
2. Search string: sequence of characters to be replaced.
3. Replacement string: sequence of characters to replace the search string.

Strings can be passed as an expression.

Usage examples	Usage result
`replace(Name, 'UserA', 'UserB')`	Returns a string from the `Name` event field in which all occurrences of `UserA` are replaced with `UserB`.
`replace($otherVariable, ' text ', '_text_')`	Returns a string from `otherVariable` in which all occurrences of `' text'` are replaced with `'_text_'`.

regexp_replace function

Replaces a sequence of characters that match a regular expression with a sequence of characters and regular expression capturing groups. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.
2. Search string: regular expression.
3. Replacement string: sequence of characters to replace the search string, and IDs of the regular expression capturing groups. A string can be passed as an expression.

Strings can be passed as a string, field name or variable. Unnamed capturing groups can be used.

In regular expressions used in variable functions, each backslash character must be additionally escaped. For example, `^example\\\\` must be used instead of the regular expression `^example\\`.

Usage examples	Usage result
`regexp_replace(SourceAddress, '([0-9]{1,3}).([0-9]{1,3}).([0-9]{1,3}).([0-9]{1,3})', 'newIP: $1.$2.$3.10')`	Returns a string from the `SourceAddress` event field in which the text `newIP` is inserted before the IP addresses. In addition, the last digits of the address are replaced with `10`.

regexp_capture function

Gets the result matching the regular expression condition from the original string. Supported for standard fields and extended event schema fields of the `string` type.

You must specify the parameters in the following sequence:

1. Original string.

2. Search string: regular expression.

Strings can be passed as a string, field name or variable. Unnamed capturing groups can be used.

In regular expressions used in variable functions, each backslash character must be additionally escaped. For example, `^example\\\\` must be used instead of the regular expression `^example\\`.

Usage examples	Example values	Usage result
`regexp_capture(Message, '(\\\\d{1,3}\\\\.\\\\d{1,3}\\\\.\\\\d{1,3}\\\\.\\\\d{1,3})')`	`Message = 'Access from 192.168.1.1 session 1'` `Message = 'Access from 45.45.45.45 translated address 192.168.1.1 session 1'`	`'192.168.1.1'` `'45.45.45.45'`

template function

Returns the string specified in the function, with variables replaced with their values. Variables for substitution can be passed in the following ways:

- Inside the string.

- After the string. In this case, inside the string, you must specify variables in the `{{index.<n>}}` notation, where <n> is the index of the variable passed after the string. The index is 0-based.

Usage examples
`template('Very long text with values of rule={{.DeviceCustomString1}} and {{.Name}} event fields, as well as values of {{index.0}} and {{index.1}} local variables and then {{index.2}}', $var1, $var2, $var10)`

Operations with timestamps

now function

Gets a timestamp in epoch format. Runs with no arguments.

Usage examples
`now()`

extract_from_timestamp function

Gets atomic time representations (year, month, day, hour, minute, second, day of the week) from fields and variables with time in the epoch format.

The parameters must be specified in the following sequence:

1. Event field of the timestamp type, or variable.
2. Notation of the atomic time representation. This parameter is case sensitive.

   Possible variants of atomic time notation:

   - y refers to the year in number format.
   - M refers to the month in number notation.
   - d refers to the number of the month.
   - wd refers to the day of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
   - h refers to the hour in 24-hour format.
   - m refers to the minutes.
   - s refers to the seconds.
3. (optional) Time zone notation.

   The parameter can be specified as a string (for example, `Europe/Moscow`) or as a variable (`$timezone`). The variable must be declared in the local variables of the correlation rule. The variable value must contain a string identifier of the time zone. It is also allowed to use variables derived from other variables and dictionary values.

   If this parameter is not specified, the time is calculated in UTC format.

Usage examples
`extract_from_timestamp(Timestamp, 'wd')`
`extract_from_timestamp(Timestamp, 'h')`
`extract_from_timestamp($otherVariable, 'h')`
`extract_from_timestamp(Timestamp, 'h', 'Europe/Moscow')`
`extract_from_timestamp(Timestamp, 'h', $timezone)`

Usage example: using a variable obtained from a dictionary

`$key_timezone = 'timezone'`
 	
`$timezone = dict('UEBA.Timezone', $key_timezone)`
 	
`extract_from_timestamp(EndTime, 'h', $timezone)`

In this example, the `$timezone` variable is derived from another variable and a dictionary value and is used as the `timezone` parameter.

parse_timestamp function

Converts the time from RFC3339 format (for example, `2022-05-24 00:00:00`, `2022-05-24 00:00:00+0300`) to epoch format.

Usage examples
`parse_timestamp(Message)`
`parse_timestamp($otherVariable)`

format_timestamp function

Converts the time from epoch format to RFC3339 format.

The parameters must be specified in the following sequence:

1. Event field of the timestamp type, or variable.
2. Time format notation: RFC3339.
3. (optional) Time zone notation. If this parameter is not specified, the time is calculated in UTC format.

Usage examples
`format_timestamp(Timestamp, 'RFC3339')`
`format_timestamp($otherVariable, 'RFC3339')`
`format_timestamp(Timestamp, 'RFC3339', 'Europe/Moscow')`

truncate_timestamp function

Rounds the time in epoch format. After rounding, the time is returned in epoch format. Time is rounded down.

The parameters must be specified in the following sequence:

1. Event field of the timestamp type, or variable.
2. Rounding parameter:
   - 1s rounds to the nearest second.
   - 1m rounds to the nearest minute.
   - 1h rounds to the nearest hour.
   - 24h rounds to the nearest day.
3. (optional) Time zone notation. If this parameter is not specified, the time is calculated in UTC format.

Usage examples	Examples of rounded values	Usage result
`truncate_timestamp(Timestamp, '1m')`	1654631774175 (7 June 2022, 19:56:14.175)	1654631760000 (7 June 2022, 19:56:00)
`truncate_timestamp($otherVariable, '1h')`	1654631774175 (7 June 2022, 19:56:14.175)	1654628400000 (7 June 2022, 19:00:00)
`truncate_timestamp(Timestamp, '24h', 'Europe/Moscow')`	1654631774175 (7 June 2022, 19:56:14.175)	1654560000000 (7 June 2022, 0:00:00)

time_diff function

Gets the time interval between two timestamps in epoch format.

The parameters must be specified in the following sequence:

1. Interval end time. Event field of the timestamp type, or variable.
2. Interval start time. Event field of the timestamp type, or variable.
3. Time interval notation:
   - ms refers to milliseconds.
   - s refers to seconds.
   - m refers to minutes.
   - h refers to hours.
   - d refers to days.

Usage examples
`time_diff(EndTime, StartTime, 's')`
`time_diff($otherVariable, Timestamp, 'h')`
`time_diff(Timestamp, DeviceReceiptTime, 'd')`

Mathematical operations

These are comprised of basic mathematical operations and functions.

Basic mathematical operations

Supported for integer and float fields of the extended event schema.

Operations:

- Addition
- Subtraction
- Multiplication
- Division
- Modulo division

Parentheses determine the sequence of actions

Available arguments:

- Numeric event fields
- Numeric variables
- Real numbers

  When modulo dividing, only natural numbers can be used as arguments.

Usage constraints:

- Division by zero returns zero.
- Mathematical operations on a number and a strings return the number unchanged. For example, 1 + abc returns 1.
- Integers resulting from operations are returned without a dot.

Usage examples `(Type=3; otherVariable=2; Message=text)`	Usage result
`Type + 1`	4
`$otherVariable - Type`	-1
`2 * 2.5`	5
`2 / 0`	0
`Type * Message`	0
`(Type + 2) * 2`	10
`Type % $otherVariable`	1

round function

Rounds numbers. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples `(DeviceCustomFloatingPoint1=7.75; DeviceCustomFloatingPoint2=7.5 otherVariable=7.2)`	Usage result
`round(DeviceCustomFloatingPoint1)`	8
`round(DeviceCustomFloatingPoint2)`	8
`round($otherVariable)`	7

ceil function

Rounds up numbers. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples `(DeviceCustomFloatingPoint1=7.15; otherVariable=8.2)`	Usage result
`ceil(DeviceCustomFloatingPoint1)`	8
`ceil($otherVariable)`	9

floor function

Rounds down numbers. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples `(DeviceCustomFloatingPoint1=7.15; otherVariable=8.2)`	Usage result
`floor(DeviceCustomFloatingPoint1)`	7
`floor($otherVariable)`	8

abs function

Gets the modulus of a number. Supported for integer and float fields of the extended event schema.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples `(DeviceCustomNumber1=-7; otherVariable=-2)`	Usage result
`abs(DeviceCustomFloatingPoint1)`	7
`abs($otherVariable)`	2

pow function

Exponentiates a number. Supported for integer and float fields of the extended event schema.

The parameters must be specified in the following sequence:

1. Base — real numbers.
2. Power — natural numbers.

Available arguments:

- Numeric event fields
- Numeric variables
- Numeric constants

Usage examples
`pow(DeviceCustomNumber1, DeviceCustomNumber2)`
`pow($otherVariable, DeviceCustomNumber1)`

str_join function

Join multiple strings into one using a separator. Supported for integer and float fields of the extended event schema.

The parameters must be specified in the following sequence:

1. Separator. String.
2. String1, string2, stringN. At least 2 expressions.

Usage examples	Usage result
`str_join('|', to_lower(Name), to_upper(Name), Name)`	String.

conditional function

Get one value if a condition is met and another value if the condition is not met. Supported for integer and float fields of the extended event schema.

The parameters must be specified in the following sequence:

1. Condition. String. The syntax is similar to the conditions of the Where statement in SQL. You can use the functions of the KUMA variables and references to other variables in a condition.
2. The value if the condition is met. Expression.
3. The value if the condition is not met. Expression.

Supported operators:

- `AND`
- `OR`
- `NOT`

- `=`
- `!=`
- `<`
- `<=`
- `>`
- `>=`
- `LIKE` (RE2 regular expression is used, rather than an SQL expression)
- `ILIKE` (RE2 regular expression is used, rather than an SQL expression)
- `BETWEEN`
- `IN`
- `IS NULL` (check for an empty value, such as 0 or an empty string)

Usage examples (the value depends on arguments 2 and 3)
`conditional('SourceUserName = \\'root\\' AND DestinationUserName = SourceUserName', 'match', 'no match')`
`conditional(`DestinationUserName ILIKE 'svc_.*'`, 'match', 'no match')`
`conditional(`DestinationUserName NOT LIKE 'svc_.*'`, 'match', 'no match')`

Operations for extended event schema fields

For extended event schema fields of the `string` type, the following kinds of operations are supported:

- `len` function
- `to_lower` function
- `to_upper` function
- `append` function
- `prepend` function
- `substring` function
- `tr` function
- `replace` function
- `regexp_replace` function
- `regexp_capture` function

For extended event schema fields of the integer or float type, the following kinds of operations are supported:

- Basic mathematical operations:
- `round` function
- `ceil` function
- `floor` function
- `abs` function
- `pow` function
- `str_join` function
- `conditional` function

For extended event schema fields of the `array of integers`, `array of floats`, and `array of strings` types, KUMA supports the following functions:

- Get the i-th element of the array. Example: `item(<type>.someStringArray)`.
- Get an array of values. Example: `<type>.someStringArray`. Returns `["string1", "string2", "string3"]`.
- Get the count of elements in an array. Example: `len(<type>.someStringArray)`. Returns `["string1", "string2"]`.
- Get unique elements from an array. Example: `distinct_items(<type>.someStringArray)`.
- Generate a TSV string of array elements. Example: `to_string(<type>.someStringArray)`.
- Sort the elements of the array. Example: `sort_items(<type>.someStringArray)`.

  In the examples, instead of `<type>`, you must specify the array type: NA for an array of integers, FA for an array of floats, SA for an array of strings.

For fields of the `array of integers` and `array of floats` types, the following functions are supported:

- `math_min` — returns the minimum element of an array. Example: `math_min(NA.NumberArray)`, `math_min(FA.FloatArray)`.
- `math_max` — returns the maximum element of an array Example: `math_max(NA.NumberArray)`, `math_max(FA.FloatArray)`.
- `math_avg` — returns the average value of an array. Example: `math_avg(NA.NumberArray)`, `math_avg(FA.FloatArray)`.

Declaring variables

To declare variables, they must be added to a correlator or correlation rule.

To add a global variable to an existing correlator:

1. In the main menu, go to **Monitoring** → **Resources** → **Correlators**.
2. Select the resource set of the relevant correlator.

   The Correlator Installation Wizard opens.
3. Select the **Global variables** step of the Installation Wizard.
4. Click the **Add variable** button and specify the following parameters:
5. In the **Variable** window, enter the name of the variable.

   Variable naming requirements

   - Must be unique within the correlator.
   - Must contain 1 to 128 Unicode characters.
   - Must not begin with the character $.
   - Must be written in camelCase or CamelCase.

   - In the **Value** window, enter the variable function.

     Description of variable functions.

   Multiple variables can be added. Added variables can be edited or deleted by using the × icon.
6. Select the **Setup validation** step of the Installation Wizard and click **Save**.

A global variable is added to the correlator. It can be queried like an event field by inserting the $ character in front of the variable name. The variable will be used for correlation after restarting the correlator service.

To add a local variable to an existing correlation rule:

1. In the main menu, go to **Monitoring** → **Resources** → **Correlation rules**.
2. Select the relevant correlation rule.

   The correlation rule settings window opens. The parameters of a correlation rule can also be opened from the correlator to which it was added by proceeding to the **Correlation** step of the Installation Wizard.
3. Open the **Selectors** tab.
4. In the selector, open the **Local variables** tab, click the **Add variable** button and specify the following parameters:

   - In the **Variable** window, enter the name of the variable.

     Variable naming requirements

     - Must be unique within the correlator.
     - Must contain 1 to 128 Unicode characters.

- Must not begin with the character $.
- Must be written in camelCase or CamelCase.

- In the **Value** window, enter the variable function.

  Description of variable functions.

Multiple variables can be added. Added variables can be edited or deleted by using the ✕ icon.

For standard correlation rules, repeat this step for each selector in which you want to declare variables.

5. Click **Save**.

The local variable is added to the correlation rule. It can be queried like an event field by inserting the $ character in front of the variable name. The variable will be used for correlation after restarting the correlator service.

Added variables can be edited or deleted. If the correlation rule queries an undeclared variable (for example, if its name has been changed), an empty string is returned.

If you change the name of a variable, you will need to manually change the name of this variable in all correlation rules where you have used it.

Predefined correlation rules

The OSMP distribution kit includes correlation rules listed in the table below.

Predefined correlation rules

Correlation rule name	Description
[OOTB] KATA alert	Used for enriching KATA events.
[OOTB] Successful Bruteforce	Triggers when a successful authentication attempt is detected after multiple unsuccessful authentication attempts. This rule works based on the events of the sshd daemon.
[OOTB][AD] Account created and deleted within a short period	Detects instances of creation and subsequent deletion of accounts on Microsoft Windows hosts.
[OOTB][AD] An account failed to log on from different hosts	Detects multiple unsuccessful attempts to authenticate on different hosts.
[OOTB][AD] Granted TGS without TGT (Golden Ticket)	Detects suspected Golden Ticket type attacks. This rule works based on Microsoft Windows events.
[OOTB][AD][Technical] 4768. TGT Requested	The technical rule used to populate the active list is [OOTB][AD] List of requested TGT. EventID 4768. This rule works based on Microsoft Windows events.
[OOTB][AD] Membership of sensitive group was modified	Works based on Microsoft Windows events.
[OOTB][AD] Multiple accounts failed to log on from the same host	Triggers after multiple failed authentication attempts are detected on the same host from different accounts.
[OOTB][AD] Possible Kerberoasting attack	Detects suspected Kerberoasting type attacks. This rule works based on Microsoft Windows events.
[OOTB][AD] Successful authentication with the same account on multiple hosts	Detects connections to different hosts under the same account. This rule works based on Microsoft Windows events.
[OOTB][AD] The account added and deleted from the group in a short period	Detects the addition of a user to a group and subsequent removal. This rule works based on Microsoft Windows events.
[OOTB][Net] Possible port scan	Detects suspected port scans. This rule works based on Netflow, Ipfix events.
[OOTB] EDR IOA alert	Monitors EDR events and creates an alert when detects an EDR event that has a sign of a triggered IOA rule and marked as an alert.

[OOTB] EDR IOA SB alert	Monitors EDR events and creates an alert when detects an EDR event that was received from Auto Sandbox Scan.
[OOTB] OSMP Scanner detects	Monitors security audit events generated by the IRP component and creates an alert when an event of one of the following types is received: "YARA Scan Host", "Sandbox Detect", "Antimalware Detect", "IOC Scan Host", "YARA Scan Server".
[OOTB] KEDR event monitoring	Monitors audit events and creates an alert when an event source monitoring policy triggers.
[OOTB] Multiple login attempts to OSMP console using the same account	Detects attempts to log in to the OSMP console using the same account and generates an alert if three unsuccessful authentication attempts occur.

MITRE ATT&CK matrix coverage

If you want to assess the coverage of the MITRE ATT&CK matrix by your correlation rules:

1. Download the list of MITRE techniques from the official MITRE ATT&CK repository and import it into KUMA.
2. Map MITRE techniques to correlation rules.
3. Export correlation rules to MITRE ATT&CK Navigator.

As a result, you can visually assess the coverage of the MITRE ATT&CK matrix.

Importing the list of MITRE techniques

Only a user with the Main administrator role can import the list of MITRE techniques.

To import the list of MITRE ATT&CK techniques:

1. Download the list of MITRE ATT&CK techniques from the GitHub portal.

   KUMA supports only the MITRE ATT&CK technique list version 14.1.

2. In the main menu, go to **Settings** → **KUMA** → **Open KUMA settings**.

   This opens the KUMA settings window.

3. Under **Other**, click **Common**.
4. In the **MITRE technique list settings**, click **Import from file**.

   The file selection window opens.

5. Select the downloaded MITRE ATT&CK technique list and click **Open**.

The list of MITRE ATT&CK techniques is imported into KUMA. You can see the list of imported techniques and the version of the MITRE ATT&CK technique list by clicking **View list**.

Mapping MITRE techniques to correlation rules

To map MITRE ATT&CK techniques to correlation rules:

1. In the main menu, go to **Monitoring** → **Resources** → **Correlation rules**.
2. Click the name of the correlation rule to open the correlation rule editing window.

   This opens the correlation rule editing window.
3. On the **General** tab, clicking the **MITRE techniques** field opens a list of available techniques. For the convenience of searching, a filter is provided, in which you can enter the name of a technique or the ID of a technique or tactic. One or more MITRE ATT&CK techniques are available for linking to a correlation rule.
4. Click **Save**.

The MITRE ATT&CK techniques are mapped to the correlation rule. In the KUMA Console, in the **Monitoring** → **Resources** → **Correlation rules** section, the **MITRE techniques** column of the edited rule displays the ID of the selected technique, and when you hover over the ID, the full name of the technique is displayed, including the ID of the technique and tactic.

Exporting correlation rules to MITRE ATT&CK Navigator

To export correlation rules with mapped MITRE techniques to MITRE ATT&CK Navigator:

1. In the main menu, go to **Monitoring** → **Resources** → **Correlation rules**.
2. Click the ellipsis button (**...**) in the upper-right corner.
3. In the drop-down list, click **Export to MITRE ATT&CK Navigator**.
4. In the window that opens, select the correlation rules that you want to export.
5. Click **OK**.

   A file with exported rules is downloaded to your computer.
6. Upload the file from your computer to MITRE ATT&CK Navigator to assess the coverage of the MITRE ATT&CK matrix.

You can assess the coverage of the MITRE ATT&CK matrix.

Filters

Filters let you select events based on specified conditions.

The collector service uses filters to select events that you want to send to KUMA. That is, an event that matches the filter condition is sent to KUMA for further processing.

Filters can be used in the following KUMA services and features:

- Collector.
- Correlator.
- Storage.
- KUMA agents.
- Correlation rules.
- Enrichment rules.
- Aggregation rules.
- Destinations.
- Response rules.
- Segmentation rules.

You can use standalone filters or built-in filters that are stored in the service or resource where they were created.

For these resources, you can enable the display of control characters in all input fields except the **Description** field.

Available settings for filters:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters. Inline filters are created in other resources or services and do not have names.
- **Tenant** (required)—name of the tenant that owns the resource.
- Description—up to 4,000 Unicode characters describing the filter.
- The **Conditions** group of settings lets you formulate filtering criteria by creating filter conditions and groups of filters, or by adding existing filters.

  To create filtering criteria, you can use *builder mode* or *source code mode*. The builder mode is used by default.

  In builder mode, you can create or edit filter criteria by selecting filter conditions and operators from drop-down lists.

  In source code mode, you can use text commands to create and edit search queries.

  You can freely switch between modes when creating filtering criteria. To switch to source code mode, click the **Code** button. When switching between modes, the created condition filters are preserved. If the filter code is not displayed on the **Code** tab after linking the created filter to the resource, go to the **Builder** tab and then go back to the **Code** tab. The filter code is displayed.

Creating conditions in builder mode

You can create filtering criteria in builder mode by clicking the following buttons:

- **Add condition** adds a string with fields for defining a condition.
- **Add group** adds a group of filters. Group operators can be switched between **AND**, **OR**, and **NOT**. You can add groups, conditions, and existing filters to groups of filters. Conditions placed in the **NOT** subgroup are combined with the **AND** operator.

To replace an operator in the created condition, click the operator that you want to replace and select the new operator from the drop-down list.

To delete an operator in the created condition, click the operator that you want to delete and press **`Backspace`**.

To alter the sequence of filter conditions, click ⣿ and drag and drop the condition to the new location.

Conditions, groups, and filters can be deleted by clicking the × button.

Settings of conditions:

- **When** (required)—in this drop-down list, you can specify whether or not to use the inverted function of the operator.
- **Left operand** and **Right operand** (required)—used to specify the values that the operator will process. The available types depend on the selected operator.

  Operands of filters

  - In the **Event fields** section, you can specify the event field that you want to use as a filter operand.
  - In the **Active lists** section, you can specify an active list or field of an active list that you want to use as an operand of the filter. When selecting an active list, you must specify one or more event fields that are used to create an active list entry and act as the key of the active list entry. To finish specifying event fields, press **`Ctrl`**/**`Command`**+**`F1`**.

    If you have not specified the **inActiveList** operator, you need to specify the name of the active list field that you want to use as a filter operand.
  - In the **Context tables** section, you can specify the value of the context table that you want to use as the filter operand. When selecting a context table, you must specify an event field:
    - **context table name** (required) is the context table that you want to use.
    - **key fields** (required) are event fields or local variables that are used to create a context table record and serve as the key for the context table record.
    - **field** is the name of the context table field from which you want to get the value of the operand.
    - **index** is the index of the list field of the table from which you want to get the value of the operand.

- **Dictionary** is a value from the dictionary resource that you want to assign to the operand. Advanced settings:
  - **dictionary** (required) is the dictionary that you want to use.
  - **key fields** (required) are event fields that are used to generate the key of the dictionary value.
- **Constant** is a user-defined value that you want to assign to the operand. Advanced settings:
  - **value** (required) is the constant that you want to assign to the operand.
- **Table** specifies user-defined values that you want to assign to the operand. Advanced settings:
  - **dictionary** (required) is the type of the dictionary. You need to select the **Table** dictionary type.
  - **key fields** (required) are event fields that are used to generate the key of the dictionary value.
- **List** specifies user-defined values that you want to assign to the operand. Advanced settings:
  - **value** (required) are the constants that you want to assign to the operand. When you type the value in the field and press **ENTER**, the value is added to the list and you can enter a new value.
- **TI** specifies the settings for reading CyberTrace threat intelligence (TI) data from the events. Advanced settings:
  - **stream** (required) is the CyberTrace threat category.
  - **key fields** (required) is the event field with CyberTrace threat indicators.
  - **field** (required) is the field of the CyberTrace feed with threat indicators.

- **Operator** (required)—used to select the condition operator.

  In this drop-down list, you can select the **do not match case** check box if the operator should ignore the case of values. This check box is ignored if the **inSubnet**, **inActiveList**, **inCategory**, **InActiveDirectoryGroup, hasBit**, **inDictionary** operators are selected. This check box is cleared by default.

  Filter operators

  - **=**—the left operand equals the right operand.
  - **<**—the left operand is less than the right operand.
  - **<=**—the left operand is less than or equal to the right operand.
  - **>**—the left operand is greater than the right operand.
  - **>=**—the left operand is greater than or equal to the right operand.
  - **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet).
  - **contains**—the left operand contains values of the right operand.
  - **startsWith**—the left operand starts with one of the values of the right operand.

- **endsWith**—the left operand ends with one of the values of the right operand.
- **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used.
- **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list).

  The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list.

  If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`.
- **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand.

  If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability.
- **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list.
- **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields.
- **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand.
- **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand.
- **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators.
- **inContextTable**—presence of the entry in the specified context table.
- **intersect**—presence in the left operand of the list items specified in the right operand.

The available operand kinds depends on whether the operand is left (L) or right (R).

Available operand kinds for left (L) and right (R) operands

Operator	Event field type	Active list type	Dictionary type	Context table type	Table type	TI type	Constant type	List type
=	L,R	L,R	L,R	L,R	L,R	L,R	R	R
>	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	—
>=	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	—
<	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	—
<=	L,R	L,R	L,R	L,R (only when looking up a table value by index)	L,R	L	R	—
inSubnet	L,R	L,R	L,R	L,R	L,R	L,R	R	R
contains	L,R	L,R	L,R	L,R	L,R	L,R	R	R

startsWith	L,R	L,R	L,R	L,R	L,R	L,R	R	R
endsWith	L,R	L,R	L,R	L,R	L,R	L,R	R	R
match	L	L	L	L	L	L	R	R
hasVulnerability	L	L	L	L	L	—	—	—
hasBit	L	L	L	L	L	—	R	R
inActiveList	—	—	—	—	—	—	—	—
inDictionary	—	—	—	—	—	—	—	—
inCategory	L	L	L	L	L	—	R	R
inContextTable	—	—	—	—	—	—	—	—
inActiveDirectoryGroup	L	L	L	L	L	—	R	R
TIDetect	—	—	—	—	—	—	—	—

You can use hotkeys when managing filters. Hotkeys are described in the table below.

Hotkeys and their functions

Key	Function
e	Invokes a filter by the event field
d	Invokes a filter by the dictionary field
a	Invokes a filter by the active list field
c	Invokes a filter by the context table field
t	Invokes a filter by the table field
f	Invokes a filter
t+i	Invokes a filter using TI
Ctrl+Enter	Finish editing a condition

The usage of extended event schema fields `string`, `number`, or `float` types is the same as the usage of fields of the KUMA event schema.

When using filters with extended event schema fields of the `Array of strings`, `Array of numbers`, and `Array of floats` types, you can use the following operations:

- The `contains` operation returns `True` if the specified substring is present in the array, otherwise it returns `False`.
- The `match` operation matches the string against a regular expression.
- The `intersec` operation.

When using filters with extended event schema fields of the `Array of numbers` and `Array of floats` types, you can use the following comparison operations: `<`, `>`, `=`, `>=`, `<=`.

If you want to reference a specific element of an array in the filter, you can use the following syntax: `NA.<array name>.<index of the element>`

Arrays are 0-based.

Example:

`NA.ArrayOne.0` — access to the first element of the ArrayOne array of integers.

`FA.ArrayTwo.2` — access to the third element of the ArrayTwo array of floats.

Creating conditions in source code mode

The code editor mode allows you to quickly edit conditions, select and copy blocks of code.

On the right side of the builder, you can find the navigator, which lets you to navigate the filter code.

Line wrapping is performed automatically at **AND**, **OR**, **NOT** logical operators, or at commas that delimit the items in the list of values.

Names of resources used in the filter are automatically specified. Fields containing the names of linked resources cannot be edited. The names of shared resource categories are not displayed in the filter if you do not have the **Access to shared resources** role.

The filters listed in the table below are included in the OSMP distribution kit.

Predefined filters

Filter name	Description
[OOTB][AD] A member was added to a security-enabled global group (4728)	Selects events of adding a user to an Active Directory security-enabled global group.
[OOTB][AD] A member was added to a security-enabled universal group (4756)	Selects events of adding a user to an Active Directory security-enabled universal group.
[OOTB][AD] A member was removed from a security-enabled global group (4729)	Selects events of removing a user from an Active Directory security-enabled global group.
[OOTB][AD] A member was removed from a security-enabled universal group (4757)	Selects events of removing a user from an Active Directory security-enabled universal group.
[OOTB][AD] Account Created	Selects Windows user account creation events.
[OOTB][AD] Account Deleted	Selects Windows user account deletion events.
[OOTB][AD] An account failed to log on (4625)	Selects Windows logon failure events.
[OOTB][AD] Successful Kerberos authentication (4624, 4768, 4769, 4770)	Selects successful Windows logon events and events with IDs 4769, 4770 that are logged on domain controllers.
[OOTB][AD][Technical] 4768. TGT Requested	Selects Microsoft Windows events with ID 4768.
[OOTB][Net] Possible port scan	Selects events that may indicate a port scan.
[OOTB][SSH] Accepted Password	Selects events of successful SSH connections with a password.
[OOTB][SSH] Failed Password	Selects attempts to connect over SSH with a password.

Active lists

The active list is a bucket for data that is used by KUMA correlators for analyzing events according to the correlation rules.

For example, for a list of IP addresses with a bad reputation, you can:

1. Create a correlation rule of the operational type and add these IP addresses to the active list.

2. Create a correlation rule of the standard type and specify the active list as filtering criteria.

3. Create a correlator with this rule.

   In this case, KUMA selects all events that contain the IP addresses in the active list and creates a correlation event.

You can fill active lists automatically using correlation rules of the simple type or import a file that contains data for the active list.

You can add, copy, or delete active lists.

Active lists can be used in the following KUMA services and features:

- Correlation rules.
- Dashboard.

The same active list can be used by different correlators. However, a separate entity of the active list is created for each correlator. Therefore, the contents of the active lists used by different correlators differ even if the active lists have the same names and IDs.

Only data based on correlation rules of the correlator are added to the active list.

You can add, edit, duplicate, delete, and export records in the active correlator sheet.

During the correlation process, when entries are deleted from active lists, service events are generated in the correlators. These events only exist in the correlators, and they are not redirected to other destinations. Correlation rules can be configured to track these events so that they can be used to identify threats. Service event fields for deleting an entry from the active list are described below.

Event field	Value or comment
`ID`	Event identifier
`Timestamp`	Time when the expired entry was deleted
`Name`	`"active list record expired"`
`DeviceVendor`	`"Kaspersky"`
`DeviceProduct`	`"KUMA"`
`ServiceID`	Correlator ID
`ServiceName`	Correlator name
`DeviceExternalID`	Active list ID
`DevicePayloadID`	Key of the expired entry
`BaseEventCount`	Number of deleted entry updates increased by one
`S.<active list field>`	Dropped-out entry of the active list in the following format: S.<active list field> = <value of active list field>

Viewing the table of active lists

To view the table of correlator active lists:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

The table contains the following data:

- **Name**—the name of the correlator list.
- **Records**—the number of record the active list contains.
- **Size on disk**—the size of the active list.
- **Directory**—the path to the active list on the KUMA Core server.

Adding active list

To add active list:

1. In the main menu, go to **Monitoring** → **Resources** → **Active lists**.
2. Click the **Add active list** button.
3. Do the following:

   a. In the **Name** field, enter a name for the active list.

   b. In the **Tenant** drop-down list, select the tenant that owns the resource.

   c. In the **TTL** field, specify time the record added to the active list is stored in it.
   When the specified time expires, the record is deleted. The time is specified in seconds.
   The default value is 0. If the value of the field is 0, the record is retained for 36,000 days (roughly 100 years).

   d. In the **Description** field, provide any additional information.
   You can use up to 4,000 Unicode characters.
   This field is optional.

4. Click the **Save** button.

   The active list is added.

Viewing the settings of an active list

To view the settings of an active list:

1. In the main menu, go to **Monitoring** → **Resources** → **Active lists**.
2. In the **Name** column, select the active list whose settings you want to view.

This opens the active list settings window. It displays the following information:

- **ID**—identifier selected Active list.
- **Name**—unique name of the resource.
- **Tenant**—the name of the tenant that owns the resource.
- **TTL**—the record added to the active list is stored in it for this time. This value is specified in seconds.
- **Description**—any additional information about the resource.

Changing the settings of an active list

To change the settings of an active list:

1. In the main menu, go to **Monitoring** → **Resources** → **Active lists**.
2. In the **Name** column, select the active list whose settings you want to change.
3. Specify the values of the following parameters:
   - **Name**—unique name of the resource.
   - **TTL**—the record added to the active list is stored in it for this time. This value is specified in seconds.
     If the field is set to 0, the record is stored indefinitely.
   - **Description**—any additional information about the resource.

   The **ID** and **Tenant** fields are not editable.

Duplicating the settings of an active list

To copy an active list:

1. In the main menu, go to **Monitoring** → **Resources** → **Active lists**.
2. Select the check box next to the active lists you want to copy.
3. Click **Duplicate**.
4. Specify the necessary settings.
5. Click the **Save** button.

The active list is copied.

Deleting an active list

To delete an active list:

1. In the main menu, go to **Monitoring** → **Resources** → **Active lists**.
2. Select the check boxes next to the active lists you want to delete.

   To delete all lists, select the check box next to the **Name** column.

   > At least one check box must be selected.

3. Click the **Delete** button.
4. Click **OK**.

The active lists are deleted.

Viewing records in the active list

To view the records in the active list:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. In the **Name** column, select the desired active list.

A table of records for the selected list is opened.

The table contains the following data:

- **Key** – the value of the record key.
- **Record repetitions** – total number of times the record was mentioned in events and identical records were downloaded when importing active lists to KUMA.
- **Expiration date** – date and time when the record must be deleted.

  If the **TTL** field had the value of 0 when the active list was created, the records of this active list are retained for 36,000 days (roughly 100 years).
- **Created** – the time when the active list was created.
- **Updated** – the time when the active list was last updated.

Searching for records in the active list

To find a record in the active list:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.
   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.
   A window with the records for the selected list is opened.
6. In the **Search** field, enter the record key value or several characters from the key.

The table of records of the active list displays only the records with the key containing the entered characters.

Adding a record to an active list

To add a record to the active list:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the required correlator.
4. Click the **Go to active lists** button.
   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.
   A window with the records for the selected list is opened.
6. Click **Add**.
   The **Create record** window opens.
7. Specify the values of the following parameters:

   a. In the **Key** field, enter the name of the record.
      You can specify several values separated by the "|" character.

      > The **Key** field cannot be empty. If the field is not filled in, KUMA returns an error when trying to save the changes.

b. In the **Value** field, specify the values for fields in the **Field** column.

KUMA takes field names from the correlation rules with which the active list is associated. These names are not editable. You can delete these fields if necessary.

c. Click the **Add new element** button to add more values.

d. In the **Field** column, specify the field name.

The name must meet the following requirements:

- To be unique.
- Do not contain tab characters.
- Do not contain special characters except for the underscore character.
- The maximum number of characters is 128.

> The name must not begin with an underscore and contain only numbers.

e. In the **Value** column, specify the value for this field.

It must meet the following requirements:

- Do not contain tab characters.
- Do not contain special characters except for the underscore character.
- The maximum number of characters is 1024.

This field is optional.

8. Click the **Save** button.

The record is added. After saving, the records in the active list are sorted in alphabet order.

Duplicating records in the active list

To duplicate a record in the active list:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.

   A window with the records for the selected list is opened.
6. Select the check boxes next to the record you want to copy.

7. Click **Duplicate**.

8. Specify the necessary settings.

> The **Key** field cannot be empty. If the field is not filled in, KUMA returns an error when trying to save the changes.

> Editing the field names in the **Field** column is not available for the records that have been added to the active list before. You can change the names only for records added at the time of editing. The name must not begin with an underscore and contain only numbers.

9. Click the **Save** button.

The record is copied. After saving, the records in the active list are sorted in alphabet order.

Changing a record in the active list

To edit a record in the active list:

1. In the main menu, go to **Monitoring** → **Resources**.

2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the correlator for which you want to view the active list.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. In the **Name** column, select the desired active list.

   A window with the records for the selected list is opened.

6. Click the record name in the **Key** column.

7. Specify the required values.

8. Click the **Save** button.

The record is overwritten. After saving, the records in the active list are sorted in alphabet order.

Restrictions when editing a record:

- The record name is not editable. You can change it by importing the same data with a different name.

- Editing the field names in the **Field** column is not available for the records that have been added to the active list before. You can change the names only for records added at the time of editing. The name must not begin with an underscore and contain only numbers.

- The values in the **Value** column must meet the following requirements:
  - Do not contain Cyrillic characters.
  - Do not contain spaces or tabs.
  - Do not contain special characters except for the underscore character.
  - The maximum number of characters is 128.

Deleting records from the active list

To delete records from the active list:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.
   The **Correlator active lists** table is displayed.
5. In the **Name** column, select the desired active list.
   A window with the records for the selected list is opened.
6. Select the check boxes next to the records you want to delete.
   To delete all records, select the check box next to the **Key** column.

   > At least one check box must be selected.

7. Click the **Delete** button.
8. Click **OK**.

The records will be deleted.

Import data to an active list

To import active list:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. Select the check box next to the correlator for which you want to view the active list.
4. Click the **Go to active lists** button.
   The **Correlator active lists** table is displayed.

5. Point the mouse over the row with the desired active list.

6. Click ... to the left of the active list name.

7. Select **Import**.

   The active list import window opens.

8. In the **File** field select the file you want to import.

9. In the **Format** drop-down list select the format of the file:

   - **csv**
   - **tsv**
   - **internal**

10. Under **Key field**, enter the name of the column containing the active list record keys.

11. Click the **Import** button.

The data from the file is imported into the active list. The records included in the list before are saved.

> Data imported from a file is not checked for invalid characters. If you use this data in widgets, widgets are displayed incorrectly if invalid characters are present in the data.

Exporting data from the active list

To export active list:

1. In the main menu, go to **Monitoring** → **Resources**.

2. In the **Services** section, click the **Active services** button.

3. Select the check box next to the correlator for which you want to view the active list.

4. Click the **Go to active lists** button.

   The **Correlator active lists** table is displayed.

5. Point the mouse over the row with the desired active list.

6. Click ... to the left of the desired active list.

7. Click the **Export** button.

The active list is downloaded in the JSON format using your browsers settings. The name of the downloaded file reflects the name of active list.

Predefined active lists

The active lists listed in the table below are included in the OSMP distribution kit.

Predefined active lists

Active list name	Description
[OOTB][AD] End-users tech support accounts	This active list is used as a filter for the [OOTB][AD] Successful authentication with same user account on multiple hosts correlation rule. Accounts of technical support staff may be added to the active list. Records are not deleted from the active list.
[OOTB][AD] List of requested TGT. EventID 4768	This active list is populated by the [OOTB][AD][Technical] 4768. TGT Requested rule, this active list is also used in the selector of the [OOTB][AD] Granted TGS without TGT (Golden Ticket) rule. Records are removed from the list 10 hours after they are recorded.
[OOTB][AD] List of sensitive groups	This active list is used as a filter for the [OOTB][AD] Membership of sensitive group was modified correlation rule. Critical domain groups, whose membership must be monitored, can be added to the active list. Records are not deleted from the active list.
[OOTB][Linux] CompromisedHosts	This active list is populated by the [OOTB] Successful Bruteforce by potentially compromised Linux hosts rule. Records are removed from the list 24 hours after they are recorded.

Dictionaries

Description of parameters

Dictionaries are resources storing data that can be used by other KUMA resources and services.

Dictionaries can be used in the following KUMA services and features:

- Collector.
- Correlation rules.
- Normalizers.

Available settings:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Description**—up to 4,000 Unicode characters describing the resource.
- **Type** (required)—type of dictionary. The selected type determines the format of the data that the dictionary can contain:
  - You can add key-value pairs to the **Dictionary** type.

    It is not recommended to add more than 50,000 entries to dictionaries of this type.

> When adding lines with the same keys to the dictionary, each new line will overwrite the existing line with the same key. This means that only one line will be added to the dictionary.

  - Data in the form of complex tables can be added to the **Table** type. You can interact with this type of dictionary by using the REST API.

- **Values** settings block—contains a table of dictionary data:

  For the **Dictionary** type, this block displays a list of **Key**—**Value** pairs. You can click the + button to add rows to the table. You can delete rows by clicking the × button that appears when you hover your mouse cursor over a row. In the **Key** field, you may optionally specify a unique value: up to 128 Unicode characters, the first character may not be $. In the **Value** field, you may optionally specify a value: up to 255 Unicode characters, the first character may not be $. You may add one or more **Key**–**Value** pairs.

  - For the **Table** type, this block displays a table containing data. You can click the + button to add rows and columns to the table. You can delete rows and columns by clicking the × buttons that are displayed when you hover your mouse cursor over a row or a column header. Column headers can be edited.

  If the dictionary contains more than 5,000 entries, they are not displayed in the KUMA Console. To view the contents of such dictionaries, the contents must be exported in CSV format. If you edit the CSV file and import it back into KUMA, the dictionary is updated.

Importing and exporting dictionaries

You can import or export dictionary data in CSV format (in UTF-8 encoding) by clicking the **Import CSV** or **Export CSV** buttons.

The format of the CSV file depends on the dictionary type:

- **Dictionary** type:

  ```
  {KEY},{VALUE}\n
  ```

- **Table** type:

  ```
  {Column header 1}, {Column header N}, {Column header N+1}\n
  {Key1}, {ValueN}, {ValueN+1}\n
  {Key2}, {ValueN}, {ValueN+1}\n
  ```

  The keys must be unique for both the CSV file and the dictionary. In tables, the keys are specified in the first column. Keys must contain 1 to 128 Unicode characters.

  Values must contain 0 to 256 Unicode characters.

During an import, the contents of the dictionary are overwritten by the imported file. When imported into the dictionary, the resource name is also changed to reflect the name of the imported file.

If the key or value contains comma or quotation mark characters (, and "), they are enclosed in quotation marks (") when exported. Also, quotation mark character (") is shielded with additional quotation mark (").

If incorrect lines are detected in the imported file (for example, invalid separators), these lines will be ignored during import into the dictionary, and the import process will be interrupted during import into the table.

Interacting with dictionaries via API

You can use the REST API to read the contents of **Table**-type dictionaries. You can also modify them even if these resources are being used by active services. This lets you, for instance, configure enrichment of events with data from dynamically changing tables exported from third-party applications.

Predefined dictionaries

The dictionaries listed in the table below are included in the OSMP distribution kit.

Predefined dictionaries

Dictionary name	Type	Description
[OOTB] Ahnlab. Severity	dictionary	Contains a table of correspondence between a priority ID and its name.
[OOTB] Ahnlab. SeverityOperational	dictionary	Contains values of the SeverityOperational parameter and a corresponding description.
[OOTB] Ahnlab. VendorAction	dictionary	Contains a table of correspondence between the ID of the operation being performed and its name.
[OOTB] Cisco ISE Message Codes	dictionary	Contains Cisco ISE event codes and their corresponding names.
[OOTB] DNS. Opcodes	dictionary	Contains a table of correspondence between decimal opcodes of DNS operations and their IANA-registered descriptions.
[OOTB] IANAProtocolNumbers	dictionary	Contains the port numbers of transport protocols (TCP, UDP) and their corresponding service names, registered by IANA.
[OOTB] Juniper - JUNOS	dictionary	Contains JUNOS event IDs and their corresponding descriptions.
[OOTB] KEDR. AccountType	dictionary	Contains the ID of the user account type and its corresponding type name.
[OOTB] KEDR. FileAttributes	dictionary	Contains IDs of file attributes stored by the file system and their corresponding descriptions.
[OOTB] KEDR. FileOperationType	dictionary	Contains IDs of file operations from the KATA API and their corresponding operation names.
[OOTB] KEDR. FileType	dictionary	Contains modified file IDs from the KATA API and their corresponding file type descriptions.
[OOTB] KEDR. IntegrityLevel	dictionary	Contains the SIDs of the Microsoft Windows INTEGRITY LEVEL parameter and their corresponding descriptions.
[OOTB] KEDR. RegistryOperationType	dictionary	Contains IDs of registry operations from the KATA API and their corresponding values.
[OOTB] Linux. Sycall types	dictionary	Contains Linux call IDs and their corresponding names.
[OOTB] MariaDB Error Codes	dictionary	The dictionary contains MariaDB error codes and is used by the [OOTB] MariaDB Audit Plugin syslog normalizer to enrich events.
[OOTB] Microsoft SQL Server codes	dictionary	Contains MS SQL Server error IDs and their corresponding descriptions.
[OOTB] MS DHCP Event IDs Description	dictionary	Contains Microsoft Windows DHCP server event IDs and their corresponding descriptions.
[OOTB] S-Terra. Dictionary MSG ID to Name	dictionary	Contains IDs of S-Terra device events and their corresponding event names.
[OOTB] S-Terra. MSG_ID to Severity	dictionary	Contains IDs of S-Terra device events and their corresponding Severity values.
[OOTB] Syslog Priority To Facility and Severity	table	The table contains the **Priority** values and the corresponding **Facility and Severity** field values.
[OOTB] VipNet Coordinator Syslog Direction	dictionary	Contains direction IDs (sequences of special characters) used in ViPNet Coordinator to designate a direction, and their corresponding values.
[OOTB] Wallix EventClassId - DeviceAction	dictionary	Contains Wallix AdminBastion event IDs and their corresponding descriptions.
[OOTB] Windows.Codes (4738)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4738 and their corresponding names.

[OOTB] Windows.Codes (4719)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4719 and their corresponding names.
[OOTB] Windows.Codes (4663)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4663 and their corresponding names.
[OOTB] Windows.Codes (4662)	dictionary	Contains operation codes present in the MS Windows audit event with ID 4662 and their corresponding names.
[OOTB] Windows. EventIDs and Event Names mapping	dictionary	Contains Windows event IDs and their corresponding event names.
[OOTB] Windows. FailureCodes (4625)	dictionary	Contains IDs from the **Failure Information\Status** and **Failure Information\Sub Status** fields of Microsoft Windows event 4625 and their corresponding descriptions.
[OOTB] Windows. ImpersonationLevels (4624)	dictionary	Contains IDs from the **Impersonation level** field of Microsoft Windows event 4624 and their corresponding descriptions.
[OOTB] Windows. KRB ResultCodes	dictionary	Contains Kerberos v5 error codes and their corresponding descriptions.
[OOTB] Windows. LogonTypes (Windows all events)	dictionary	Contains IDs of user logon types and their corresponding names.
[OOTB] Windows_Terminal Server. EventIDs and Event Names mapping	dictionary	Contains Microsoft Terminal Server event IDs and their corresponding names.
[OOTB] Windows. Validate Cred. Error Codes	dictionary	Contains IDs of user logon types and their corresponding names.
[OOTB] ViPNet Coordinator Syslog Direction	dictionary	Contains direction IDs (sequences of special characters) used in ViPNet Coordinator to designate a direction, and their corresponding values.
[OOTB] Syslog Priority To Facility and Severity	table	Contains the Priority values and the corresponding Facility and Severity field values.

Response rules

Response rules let you initiate automatic running of Kaspersky Security Center tasks, Threat Response actions for Kaspersky Endpoint Detection and Response, KICS for Networks, Active Directory, and running a custom script for specific events.

Automatic execution of Kaspersky Security Center tasks, Kaspersky Endpoint Detection and Response tasks, and KICS for Networks and Active Directory tasks in accordance with response rules is available when integrated with the relevant programs.

You can configure response rules under **Monitoring** → **Resources** → **Response rules**, and then select the created response rule from the drop-down list in the correlator settings. You can also configure response rules directly in the correlator settings.

Response rules for Kaspersky Security Center

You can configure response rules to automatically start tasks of anti-virus scan and updates on Kaspersky Security Center assets.

When creating and editing response rules for Kaspersky Security Center, you need to define values for the following settings.

General tab

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting, available if KUMA is integrated with Kaspersky Security Center. Response rule **osmptasks**.
Open Single Management Platform task	Required setting. Name of the Kaspersky Security Center task to run. Tasks must be created beforehand, and their names must begin with "`KUMA`". For example, `KUMA antivirus check` (not case-sensitive and without quotation marks). You can use KUMA to run the following types of Kaspersky Security Center tasks: • Update • Virus scan
Event field	Required setting. Defines the event field of the asset for which the Kaspersky Security Center task should be started. Possible values: • SourceAssetID • DestinationAssetID • DeviceAssetID
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the response rule. You can add up to 4,000 Unicode characters.

Setting	Description
Filter	Used to define the conditions for the events to be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

To send requests to Kaspersky Security Center, you must ensure that Kaspersky Security Center is available over the UDP protocol.

If a response rule is owned by the shared tenant, the displayed Kaspersky Security Center tasks that are available for selection are from the Kaspersky Security Center Server that the main tenant is connected to.

If a response rule has a selected task that is absent from the Kaspersky Security Center Server that the tenant is connected to, the task is not performed for assets of this tenant. This situation could arise when two tenants are using a common correlator, for example.

Services tab

This tab is displayed only when you edit the settings of the created response rule; on this tab, you can link the rule to correlators.

To link a response rule to a correlator, click the **Link to service** button → **Link to correlator**. This opens a window; in that window, select one or more correlators and click **OK**. The response rule is linked to the selected correlators and is applied in accordance with their configuration.

If necessary, you can unlink the response rule from a service. To do this, select the check box next to the relevant correlator, click **Delete**, and confirm the unlinking from the service.

Response rules for a custom script

You can create a script containing commands to be executed on the KUMA server when selected events are detected and configure response rules to automatically run this script. In this case, the program will run the script when it receives events that match the response rules.

The script file is stored on the server where the correlator service using the response resource is installed: /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts. The `kuma` user of this server requires the permissions to run the script.

When creating and editing response rules for a custom script, you need to define values for the following parameters.

General tab

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting. Response rule type, **script**.
Timeout	The number of seconds allotted for the script to finish. If this amount of time is exceeded, the script is terminated.
Script name	Required setting. Name of the script file. If the response resource is attached to the correlator service but there is no script file in the /opt/kaspersky/kuma/correlator/<Correlator ID>/scripts folder, the correlator will not work.
Script arguments	Arguments or event field values that must be passed to the script. If the script includes actions taken on files, you should specify the absolute path to these files. Parameters can be written with quotation marks ("). Event field names are passed in the `{{.EventField}}` format, where `EventField` is the name of the event field which value must be passed to the script. Example: `-n "\"usr\": {{.SourceUserName}}"`
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the resource. You can add up to 4,000 Unicode characters.

Setting	Description
Filter	Used to define the conditions for the events to be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when you edit the settings of the created response rule; on this tab, you can link the rule to correlators.

To link a response rule to a correlator, click the **Link to service** button → **Link to correlator**. This opens a window; in that window, select one or more correlators and click **OK**. The response rule is linked to the selected correlators and is applied in accordance with their configuration.

If necessary, you can unlink the response rule from a service. To do this, select the check box next to the relevant correlator, click **Delete**, and confirm the unlinking from the service.

Response rules for KICS for Networks

You can configure response rules to automatically trigger response actions on KICS for Networks assets. For example, you can change the asset status in KICS for Networks.

When creating and editing response rules for KICS for Networks, you need to define values for the following settings.

General tab

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting. Response rule type, **kics**.
Event field	Required setting. Specifies the event field for the asset for which response actions must be performed. Possible values: • SourceAssetID • DestinationAssetID • DeviceAssetID
KICS for Networks task	Response action to be performed when data is received that matches the filter. The following types of response actions are available: • **Change asset status to Authorized**. • **Change asset status to Unauthorized**. When a response rule is triggered, KUMA will send KICS for Networks an API request to change the status of the specified device to **Authorized** or **Unauthorized**.
Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.

Setting	Description
Description	Description of the resource. You can add up to 4,000 Unicode characters.

Setting	Description
Filter	Used to define the conditions for the events to be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when you edit the settings of the created response rule; on this tab, you can link the rule to correlators.

To link a response rule to a correlator, click the **Link to service** button → **Link to correlator**. This opens a window; in that window, select one or more correlators and click **OK**. The response rule is linked to the selected correlators and is applied in accordance with their configuration.

If necessary, you can unlink the response rule from a service. To do this, select the check box next to the relevant correlator, click **Delete**, and confirm the unlinking from the service.

Response rules for Kaspersky Endpoint Detection and Response

You can configure response rules to automatically trigger response actions on Kaspersky Endpoint Detection and Response assets. For example, you can configure automatic asset network isolation.

When creating and editing response rules for Kaspersky Endpoint Detection and Response, you need to define values for the following settings.

General tab

Response rule settings

Setting	Description
Event field	Required setting. Specifies the event field for the asset for which response actions must be performed. Possible values: • SourceAssetID • DestinationAssetID • DeviceAssetID

Setting	Description

Task type	Response action to be performed when data is received that matches the filter. The following types of response actions are available:
	• Enable network isolation. When selecting this type of response, you need to define values for the following setting:
	  • **Isolation timeout**—the number of hours during which the network isolation of an asset will be active. You can indicate from 1 to 9,999 hours. If necessary, you can **add an exclusion for network isolation**.
	  To add an exclusion for network isolation:
	  a. Click the **Add exclusion** button.
	  b. Select the direction of network traffic that must not be blocked:
	  • Inbound.
	  • Outbound.
	  • Inbound/Outbound.
	  c. In the **Asset IP** field, enter the IP address of the asset whose network traffic must not be blocked.
	  d. If you selected **Inbound** or **Outbound**, specify the connection ports in the **Remote ports** and **Local ports** fields.
	  e. If you want to add more than one exclusion, click **Add exclusion** and repeat the steps to fill in the **Traffic direction**, **Asset IP**, **Remote ports** and **Local ports** fields.
	  f. If you want to delete an exclusion, click the **Delete** button under the relevant exclusion.
	  When adding exclusions to a network isolation rule, Kaspersky Endpoint Detection and Response may incorrectly display the port values in the rule details. This does not affect application performance. For more details on viewing a network isolation rule, please refer to the *Kaspersky Anti Targeted Attack Platform Help Guide*.
	• Disable network isolation.
	• Add prevention rule. When selecting this type of response, you need to define values for the following settings:
	  • **Event fields to extract hash from**—event fields from which KUMA extracts SHA256 or MD5 hashes of files that must be prevented from running.
	  The selected event fields, as well as the values selected in **Event field**, must be added to the propagated fields of the correlation rule.
	  • **File hash #1**—SHA256 or MD5 hash of the file to be blocked.
	  At least one of the above fields must be completed.
	• Delete prevention rule.
	• Run program. When selecting this type of response, you need to define values for the following settings:
	  • **File path**—path to the file of the process that you want to start.
	  • **Command line parameters**—parameters with which you want to start the file.
	  • **Working directory**—directory in which the file is located at the time of startup.
	  When a response rule is triggered for users with the Main administrator role, the **Run program** task will be displayed in the **Task manager** section of the program console. **Scheduled task** is displayed for this task in the Created column of the **task table**. You can view task completion results.
	  All of the listed operations can be performed on assets that have Kaspersky Endpoint Agent for Windows. On assets that have Kaspersky Endpoint Agent for Linux, the program can only be started.
	  At the software level, the capability to create prevention rules and network isolation rules for assets with Kaspersky Endpoint Agent for Linux is unlimited. KUMA and Kaspersky Endpoint Detection and Response do not provide any notifications about unsuccessful application of these rules.

Workers	The number of processes that the service can run simultaneously. By default, the number of workers is the same as the number of virtual processors on the server where the service is installed.
Description	Description of the response rule. You can add up to 4,000 Unicode characters.

Setting	Description
Filter	Used to define the conditions for the events to be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand. d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list.

Setting	Description
	You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when you edit the settings of the created response rule; on this tab, you can link the rule to correlators.

To link a response rule to a correlator, click the **Link to service** button → **Link to correlator**. This opens a window; in that window, select one or more correlators and click **OK**. The response rule is linked to the selected correlators and is applied in accordance with their configuration.

If necessary, you can unlink the response rule from a service. To do this, select the check box next to the relevant correlator, click **Delete**, and confirm the unlinking from the service.

Active Directory response rules

Active Directory response rules define the actions to be applied to an account if a rule is triggered.

When creating and editing response rules using Active Directory, specify the values for the following settings.

General tab

Response rule settings

Setting	Description
Name	Required setting. Unique name of the resource. Must contain 1 to 128 Unicode characters.
Tenant	Required setting. The name of the tenant that owns the resource.
Type	Required setting. Response rule type, **Response via Active Directory**.
Account ID source	Event field from which the Active Directory account ID value is taken. Possible values: • SourceAccountID • DestinationAccountID

Setting	Description
AD command	Command that is applied to the account when the response rule is triggered. Available values: • **Add account to group** The Active Directory group to move the account from or to. In the mandatory field **Distinguished name**, you must specify the full path to the group. For example, CN = Team, OU = Groups, OU = ExchangeObjects, DC = example, DC = com. Only one group can be specified within one operation. • **Remove account from group** The Active Directory group to move the account from or to. In the mandatory field **Distinguished name**, you must specify the full path to the group. For example, CN = Team, OU = Groups, OU = ExchangeObjects, DC = example, DC = com. Only one group can be specified within one operation. • Reset account password If your Active Directory domain allows selecting the **User cannot change password** check box, resetting the user account password as a response will result in a conflict of requirements for the user account: the user will not be able to authenticate. The domain administrator will need to clear one of the check boxes for the affected user account: **User cannot change password** or **User must change password at next logon**. • Block account

Setting	Description
Filter	Used to define the conditions for the events to be processed using the response rule. You can select an existing filter from the drop-down list or **create** a new filter. **Creating a filter in resources** *To create a filter:* 1. In the **Filter** drop-down list, select **Create new**. 2. If you want to keep the filter as a separate resource, select the **Save filter** check box. In this case, you will be able to use the created filter in various services. This check box is cleared by default. 3. If you selected the **Save filter** check box, enter a name for the created filter resource in the **Name** field. Maximum length of the name: 128 Unicode characters. 4. In the **Conditions** settings block, specify the conditions that the events must meet: a. Click the **Add condition** button. b. In the **Left operand** and **Right operand** drop-down lists, specify the search parameters. Depending on the data source selected in the **Right operand** field, there may be fields of additional parameters for identifying the value to be passed to the filter. For example, when you select **active list**, you must specify the name of the active list, the entry key, and the entry key field. c. In the **operator** drop-down list, select an operator. **Filter operators** • **=**—the left operand equals the right operand. • **<**—the left operand is less than the right operand. • **<=**—the left operand is less than or equal to the right operand. • **>**—the left operand is greater than the right operand. • **>=**—the left operand is greater than or equal to the right operand. • **inSubnet**—the left operand (IP address) is in the subnet of the right operand (subnet). • **contains**—the left operand contains values of the right operand. • **startsWith**—the left operand starts with one of the values of the right operand. • **endsWith**—the left operand ends with one of the values of the right operand. • **match**—the left operand matches the regular expression of the right operand. The RE2 regular expressions are used. • **hasBit**—checks whether the left operand (string or number) contains bits whose positions are listed in the right operand (in a constant or in a list). The value to be checked is converted to binary and processed right to left. Chars are checked whose index is specified as a constant or a list. If the value being checked is a string, then an attempt is made to convert it to integer and process it in the way described above. If the string cannot be converted to a number, the filter returns `False`. • **hasVulnerability**—checks whether the left operand contains an asset with the vulnerability and vulnerability severity specified in the right operand. If you do not specify the ID and severity of the vulnerability, the filter is triggered if the asset in the event being checked has any vulnerability. • **inActiveList**—this operator has only one operand. Its values are selected in the **Key fields** field and are compared with the entries in the active list selected from the Active List drop-down list. • **inDictionary**—checks whether the specified dictionary contains an entry defined by the key composed with the concatenated values of the selected event fields. • **inCategory**—the asset in the left operand is assigned at least one of the asset categories of the right operand. • **inActiveDirectoryGroup**—the Active Directory account in the left operand belongs to one of the Active Directory groups in the right operand. • **TIDetect**—this operator is used to find events using CyberTrace Threat Intelligence (TI) data. This operator can be used only on events that have completed enrichment with data from CyberTrace Threat Intelligence. In other words, it can only be used in collectors at the destination selection stage and in correlators. • **inContextTable**—presence of the entry in the specified context table. • **intersect**—presence in the left operand of the list items specified in the right operand.

Setting	Description
	d. If you want the operator to be case-insensitive, select the **do not match case** check box. The selection of this check box does not apply to the **InSubnet**, **InActiveList**, **InCategory** or **InActiveDirectoryGroup** operators. This check box is cleared by default. e. If you want to add a negative condition, select **If not** from the **If** drop-down list. You can add multiple conditions or a group of conditions. 5. If you have added multiple conditions or groups of conditions, choose a selection condition (**and**, **or**, **not**) by clicking the **AND** button. 6. If you want to add existing filters that are selected from the **Select filter** drop-down list, click the **Add filter** button. You can view the nested filter settings by clicking the button.

Services tab

This tab is displayed only when you edit the settings of the created response rule; on this tab, you can link the rule to correlators.

To link a response rule to a correlator, click the **Link to service** button → **Link to correlator**. This opens a window; in that window, select one or more correlators and click **OK**. The response rule is linked to the selected correlators and is applied in accordance with their configuration.

If necessary, you can unlink the response rule from a service. To do this, select the check box next to the relevant correlator, click **Delete**, and confirm the unlinking from the service.

Virtual Administration Server

Virtual Administration Server (also referred to as *virtual Server*) is a component of Open Single Management Platform intended for managing anti-virus protection of the network of a client organization.

Virtual Administration Server is a particular case of a secondary Administration Server and has the following restrictions as compared with a physical Administration Server:

- Virtual Administration Server can be created only on a primary Administration Server.
- Virtual Administration Server uses the primary Administration Server database in its operation. Data backup and restoration tasks, as well as update scan and download tasks, are not supported on a virtual Administration Server.
- Virtual Server does not support creation of secondary Administration Servers (including virtual Servers).

In addition, virtual Administration Server has the following restrictions:

- In the virtual Administration Server properties window, the number of sections is limited.
- To install Kaspersky applications remotely on client devices managed by the virtual Administration Server, you must make sure that Network Agent is installed on one of the client devices, in order to ensure communication with the virtual Administration Server. Upon first connection to the virtual Administration Server, the device is automatically assigned as a distribution point, thus functioning as a connection gateway between the client devices and the virtual Administration Server.
- A virtual Server can poll the network only through distribution points.

- To restart a malfunctioning virtual Server, Open Single Management Platform restarts the primary Administration Server and all virtual Administration Servers.
- Users created on a virtual Server cannot be assigned a role on the Administration Server.

The administrator of a virtual Administration Server has all privileges on this particular virtual Server.

Connectors

Connectors are used for establishing connections between KUMA services and receiving events actively and passively.

The program has the following connector types available:

- tcp—used to receive data over TCP passively. Available for Windows and Linux agents.
- udp—used to receive data over UDP passively. Available for Windows and Linux agents.
- netflow—used to passively receive events in the NetFlow format.
- sflow—used to passively receive events in the SFlow format.
- nats-jetstream—used for communication with the NATS message broker. Available for Windows and Linux agents.
- kafka—used for communication with the Apache Kafka data bus. Available for Windows and Linux agents.
- http—used for receiving events over HTTP. Available for Windows and Linux agents.
- sql—used for selecting data from a database.

  The program supports the following types of SQL databases:

  - SQLite.
  - MSSQL.
  - MySQL.
  - PostgreSQL.
  - Cockroach.
  - Oracle.
  - Firebird.
  - ClickHouse.
- file—used to retrieve data from a text file. Available for Linux agents.
- 1c-log and 1c-xml are used to receive data from 1C logs. Available for Linux agents.

- diode—used for unidirectional data transfer in industrial ICS networks using data diodes.
- ftp—used to receive data over the File Transfer Protocol. Available for Windows and Linux agents.
- nfs—used to receive data over the Network File System protocol. Available for Windows and Linux agents.
- wmi—used to obtain data using Windows Management Instrumentation. Available for Windows agents.
- wec—used to receive data using Windows Event Forwarding (WEF) and Windows Event Collector (WEC), or local operating system logs of a Windows host. Available for Windows agents.
- snmp—used to receive data using the Simple Network Management Protocol. Available for Windows and Linux agents.
- snmp-trap—used to receive data using Simple Network Management Protocol traps (SNMP traps). Available for Windows and Linux agents.
- kata/edr—used to receive KEDR data via the API.
- vmware—used to receive VMware vCenter data via the API.
- elastic—used to receive Elasticsearch data.
- etw—used to receive extended DNS server logs.
- universalAPI—used to receive data from any REST APIs that support JSON responses. Filtering by unique ID and iteration mode are supported.

Viewing connector settings

To view connector settings:

1. In the main menu, go to **Monitoring** → **Resources** → **Connectors**.
2. In the folder structure, select the folder containing the relevant connector.
3. Select the connector whose settings you want to view.

The settings of connectors are displayed on two tabs: **Basic settings** and **Advanced settings**. For a detailed description of each connector settings, please refer to the *Connector settings* section.

Adding a connector

You can enable the display of non-printing characters for all entry fields except the **Description** field.

To add a connector:

1. In the main menu, go to **Monitoring** → **Resources** → **Connectors**.
2. In the folder structure, select the folder in which you want the connector to be located.

> Root folders correspond to tenants. To make a connector available to a specific tenant, the resource must be created in the folder of that tenant.

If the required folder is absent from the folder tree, you need to create it.

By default, added connectors are created in the **Shared** folder.

3. Click the **Add connector** button.

4. Define the settings for the selected connector type.

   The settings that you must specify for each type of connector are provided in the *Connector settings* section.

5. Click the **Save** button.

Connector settings

This section describes the settings of all connector types supported by KUMA.

Tcp type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **tcp**.
- **URL** (required)—URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
- **Auditd** is the toggle switch of the mechanism that groups auditd log event records received from a connector into a single event. Auditd only supports the \n delimiter, so if the toggle switch is enabled, the **Delimiter** field becomes unavailable. If the **Auditd** toggle switch is enabled in the agent connector, the \n delimiter must be configured in the connector of the collector to which the agent sends events.
- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), the default value is `\n`.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 1 MB, and the maximum value is 64 MB.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

- **Event buffer TTL** is the time to live of the buffer for grouping records into a single auditd event. This field is available if the **Auditd** toggle switch is enabled. The countdown starts the moment when the first event line is received, or immediately after the previous TTL expires. Possible values: 50 ms to 3000 ms. The default value is 2000 ms.

- **Transport header**—for auditd events, you must specify a regular expression to be used for identifying parts of the auditd log. You can use the default or edit it to suit your needs. The regular expression must contain the record_type_name, record_type_value, and event_sequence_number groups. If a multi-line auditd event contains a prefix, the prefix is retained for the first record, and for subsequent entries, the prefix is discarded.

  You can revert to the original value by clicking **Reset to default value**.

- **TLS mode**—TLS encryption mode using certificates in PEM x509 format:

  - **Disabled** (default)—do not use TLS encryption.

  - **Enabled**—use encryption without certificate verification.

  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.

  - **Custom PFX** – use encryption. When this option is selected, a certificate must be generated with a private key in PKCS#12 container format in an external Certificate Authority. Then the certificate must be exported from the key store and uploaded to the KUMA Console as a PFX secret. **Add PFX secret**.

    To create a PFX secret:

    1. In the **Name** field, enter the name of the PFX secret.

    2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX certificate with the private key.

    3. In the **Password** field, enter the PFX certificate security password that was set in the PFX Certificate Export Wizard.

    4. Click the **Create** button.

    The PFX secret is created and displayed in the **PFX secret** drop-down list.

  When using TLS, it is impossible to specify an IP address as a URL.

- **Compression**—you can use Snappy compression. By default, compression is **disabled**.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

udp type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **udp**.
- **URL** (required)—URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
- **Auditd** is the toggle switch of the mechanism that groups auditd log event records received from a connector into a single event. Auditd only supports the \n delimiter, so if the toggle switch is enabled, the **Delimiter** field becomes unavailable. If the **Auditd** toggle switch is enabled in the agent connector, the \n delimiter must be configured in the connector of the collector to which the agent sends events.
- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), events are not separated.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 16 KB, and the maximum value is 64 KB.
- **Number of handlers** is the number of handlers that the service can run simultaneously to process response rules in parallel. You can determine the number of handlers use the formula: (<number of CPU cores>/2) + 2.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Event buffer TTL** is the time to live of the buffer for grouping records into a single auditd event. This field is available if the **Auditd** toggle switch is enabled. The countdown starts the moment when the first event line is received, or immediately after the previous TTL expires. Possible values: 50 ms to 3000 ms. The default value is 2000 ms.
- **Transport header**—for auditd events, you must specify a regular expression to be used for identifying parts of the auditd log. You can use the default or edit it to suit your needs. The regular expression must contain the record_type_name, record_type_value, and event_sequence_number groups. If a multi-line auditd event contains a prefix, the prefix is retained for the first record, and for subsequent entries, the prefix is discarded.

  You can revert to the original value by clicking **Reset to default value**.
- **Compression**—you can use Snappy compression. By default, compression is **disabled**.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Netflow type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.

- **Type** (required)—connector type, **netflow**.
- **URL** (required)—URL that you need to connect to.
- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Buffer size** is used to set a buffer size for the connector. The default value is 16 KB, and the maximum value is 64 KB.
  - **Workers**—used to set worker count for the connector. The default value is 1.
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Sflow type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **sflow**.
- **URL** (required)—a URL that you need to connect to. Available formats: hostname:port, IPv4:port, IPv6:port, :port.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 1 MB, and the maximum value is 64 MB.
- **Workers**—used to set the amount of workers for a connector. The default value is 1.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Debug**—a toggle switch that lets you enable resource logging. By default, this toggle switch is in the **Disabled** position.

nats-jetstream type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **nats-jetstream**.
- **URL** (required)—URL that you need to connect to.
- **Topic** (required)—the topic for NATS messages. Must contain Unicode characters.
- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), events are not separated.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Buffer size** is used to set a buffer size for the connector. The default value is 16 KB, and the maximum value is 64 KB.
- **GroupID**—the GroupID parameter for NATS messages. Must contain 1 to 255 Unicode characters. The default value is `default`.
- **Workers**—used to set worker count for the connector. The default value is 1.
- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Cluster ID** is the ID of the NATS cluster.
- **TLS mode** specifies whether TLS encryption is used:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled**—use encryption without certificate verification.
  - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the folder /opt/kaspersky/kuma/core/certificates/.
  - **Custom CA**—use encryption with verification that the certificate was signed by a Certificate Authority. The secret containing the certificate is selected from the **Custom CA** drop-down list, which is displayed when this option is selected.

    Creating a certificate signed by a Certificate Authority

    You can create a CA-signed certificate on the KUMA Core server (the following command examples use OpenSSL).

    To create a certificate signed by a Certificate Authority:

    1. Generate a key to be used by the Certificate Authority, for example:

       ```
       openssl genrsa -out ca.key 2048
       ```

2. Create a certificate for the generated key, for example:

   ```
   openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt
   ```

3. Create a private key and a request to have it signed by the Certificate Authority, for example:

   ```
   openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=< common host name of KUMA server >" -out server.csr
   ```

4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:

   ```
   openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
   ```

5. Upload the created `server.crt` certificate in the KUMA Console to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.

When using TLS, it is impossible to specify an IP address as a URL.

To use KUMA certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, error x509: certificate signed by unknown authority may be returned.

- **Compression**—you can use Snappy compression. By default, compression is **disabled**.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Connector, kafka type

Connectors of the **kafka** type are used for communicating with the Apache Kafka data bus when working with Windows and Linux agents. Settings for a connector of the **kafka** type are described in the following tables.

Using a connector of the **kafka** type, KUMA can receive events in uncompressed as well as compressed form. Compression is performed using the following compression algorithms supported by **kafka**: gzip, snappy, zstd, or lz4. Events arriving in a compressed form are decompressed automatically.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **kafka**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. The following URL formats are supported: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` You can specify IPv6 addresses in the following format: `[< IPv6 address >%< interface >:< port number >`, for example, `[fe80::5054:ff:fe4d:ba0c%eth0]:4222`. You can add multiple URLs or remove an URL. KUMA does not allow saving a resource or service if the URL field contains a tab or space character. To add an URL, click the **+ Add** button. To remove an URL, click the delete ✕ icon next to it. Required setting.

Setting	Description
Authorization	Type of authorization when connecting to the URL specified in the **URL** field: • **Disabled**. This value is selected by default. • **Plain**. Authorization of this type is performed using the SASL/PLAIN mechanism, which transmits the user name and password in plain text. Typically, TLS is used to provide transmission security (encryption) in SASL/PLAIN. If this option is selected, in the **Secret** drop-down list, specify the secret containing user account credentials for authorization when connecting to the destination. You can select an existing secret or create a new secret. To create a new secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. **How to create a secret?** *To create a secret:* 1. In the **Name** field, enter the name of the secret. 2. In the **User** and **Password** fields, enter the credentials of the user account that the Agent will use to connect to the connector. 3. If necessary, enter a description of the secret in the **Description** field. 4. Click the **Create** button. The secret is added and displayed in the **Secret** drop-down list. • **PFX** means TLS encryption with a PFX secret. You must generate a certificate with a private key in PKCS#12 container format in an external Certificate Authority, export the PFX file from the key store, and upload the PFX file to the KUMA web interface as a PFX secret. If you select this value, in the **PFX secret** drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select **Create new**. If you want to edit the settings of an existing secret, click the pencil icon next to it. **How to create a PFX secret?** *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and is displayed in the **PFX secret** drop-down list. If you are using a chain of several certificates to authenticate the server, we recommend building the PFX file using OpenSSL to ensure the correct order of certificates in the chain. • **SCRAM-SHA-256** for authorization using a secure connection mechanism that uses the SHA256 hash for secure authentication and prevents data interception. • **SCRAM-SHA-512** for authorization using a secure connection mechanism that uses the SHA512 hash for secure authentication and prevents data interception. When **SCRAM-SHA-256** or **SCRAM-SHA-512** is selected as the authorization type in the **Secret** drop-down list, specify a secret with a password for authorization. You can select an existing secret or create a new secret. To create a new secret, in the **Secret** drop-down list, select **Create new**.
Topic	Subject of Kafka messages. The maximum length of a message is 255 characters. Space and tab characters are not allowed. You can use the following characters: a–z, A–Z, 0–9, ".", "_", "-". Required setting.
GroupID.	The value of the `GroupID` parameter for Kafka messages. The maximum length of the value is 255 characters. You can use the following characters: a–z, A–Z, 0–9, ".", "_", and "-".
Delimiter	The character that marks the boundary between events: • **\n** • **\t** • **\0** If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	This toggle switch enables logging for the resource. The toggle switch is turned off by default.
Number of handlers	Number of handlers that the service can run simultaneously to process response rules in parallel. The value must be a positive integer up to 999. By default, the number of handlers is 256.
Character encoding	Character encoding. The default is **UTF-8**.

TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values:
	• **Disabled** means TLS encryption is not used. This value is selected by default.
	Enabled means TLS encryption is used, but certificates are not verified.
	• **Custom CA** means TLS encryption is used with verification that the certificate was signed by a Certificate Authority. If you select this value, in the **Custom CA** drop-down list, specify a secret with a certificate signed by a certification authority. You can select an existing secret or create a new secret. To create a new secret, select **Create new**.
	If you want to edit the settings of an existing secret, click the pencil icon next to it.
	How to create a certificate signed by a Certificate Authority?
	You can create a CA-signed certificate on the KUMA Core server (the following command examples use OpenSSL).
	To create a certificate signed by a Certificate Authority:
	1. Generate a key to be used by the Certificate Authority, for example:
	`openssl genrsa -out ca.key 2048`
	2. Create a certificate for the generated key, for example:
	`openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt`
	3. Create a private key and a request to have it signed by the Certificate Authority, for example:
	`openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=< common host name of KUMA server >" -out server.csr`
	4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:
	`openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt`
	5. Upload the created `server.crt` certificate in the KUMA web interface to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.
	To use KUMA certificates on third-party devices, you must change the certificate file extension from CERT to CRT. Otherwise, you can get the `x509: certificate signed by unknown authority` error.

Size of message to fetch	Size of one message in the request, in bytes. The default value of 16 MB is applied if no value is specified or 0 is specified. It cannot exceed 2147483647 bytes.
Maximum fetch wait time	Timeout for one message in seconds. The default value of 5 seconds is applied if no value is specified or 0 is specified. Maximum possible value: 2147483647.
Connection timeout	Kafka broker connection timeout in seconds. Maximum possible value: 2147483647. The default value is 30 seconds.
Read timeout	Read operation timeout in seconds. Maximum possible value: 2147483647. The default value is 30 seconds.
Write timeout	Write operation timeout in seconds. Maximum possible value: 2147483647. The default value is 30 seconds.

Setting	Description
Group heartbeat interval	Group status update interval in seconds. The update interval cannot exceed the session time. The recommended value is 1/3 of the session time. Maximum possible value: 2147483647. The default value is 3 seconds.
Session timeout	The session time in seconds. Maximum possible value: 2147483647. The default value is 10 seconds.
Max processing time	Maximum duration of processing one message by a single thread, in milliseconds. Maximum possible value: 2147483647. The default value is 100 milliseconds.
Auto commit enabled	Enabled by default.
Auto commit interval	Autocommit interval in seconds. The default value is 1 second. Maximum possible value: 18446744073709551615. Any positive number can be specified.

kata/edr type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **kata/edr**.
- **URL** (required)—URL at which telemetry can be received from the KATA/EDR server. The URL must include the host and the port; the default port is 443. If KATA/EDR is deployed in a cluster, you can specify multiple URLs to ensure high availability of the connection.
- **Secret** (required) is a drop-down list in which you can select the secret which stores the credentials for connecting to the KATA/EDR server. You can select the secret resource from the drop-down list or create one by clicking the plus icon (+). When creating a secret, you can specify a custom certificate and private key, or automatically generate a new self-signed certificate and private key. You can change the selected secret by clicking the pencil icon (✎).
- **External ID**—ID for external systems. KUMA generates an ID in this field automatically.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
- **Character encoding**—the source character encoding setting for conversion to UTF-8. We only recommend configuring a conversion if you find invalid characters in the fields of the normalized event. Default value: not selected.
- **Number of events**—maximum number of events in one request. By default, the value set on the KATA/EDR server is used.
- **Events fetch timeout** is the time in seconds to wait for receipt of events from the KATA/EDR server. The default value is 0, which means that the value specified on the KATA/EDR server is used.
- **Client timeout** is the time in seconds to wait for a response from the KATA/EDR server. Default value: 1,800 s; displayed as 0.
- **KEDRQL filter**—filter of requests to the KATA/EDR server. For more details on the query language, please refer to the KEDR Help.

Connector, http type

Connectors of the **http** type are used for receiving events over HTTP when working with Windows and Linux agents. Settings for a connector of the **http** type are described in the following tables.

Using a connector of the **http** type, KUMA can receive events in uncompressed as well as compressed form. Events arriving in a compressed form, for example, Kaspersky Endpoint Security for Windows events, are decompressed automatically.

Basic settings tab

Setting	Description
Name	Unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
Tenant	The name of the tenant that owns the resource. Required setting.
Type	Connector type: **http**. Required setting.
Tags	Tags for resource search. Optional setting.
URL	URL that you want to connect to. You can enter a URL in one of the following formats: • `< host name >:< port number >` • `< IPv4 address >:< port number >` • `< IPv6 address >:< port number >` • `:< port number >` For data from the event source to be sent to the connector, the specified URL must end with `/input`. Required setting.
Delimiter	The character that marks the boundary between events: • `\n` • `\t` • `\0` If you do not select a value in this drop-down list, an empty value is selected by default. In this case, the connector does not split one event into several and passes it to the normalizer unchanged.
Description	Description of the resource. The maximum length of the description is 4000 Unicode characters.

Advanced settings tab

Setting	Description
Debug	This toggle switch enables logging for the resource. The toggle switch is turned off by default.
Character encoding	Character encoding. The default is **UTF-8**.

Setting	Description
TLS mode	TLS encryption mode. When using TLS encryption, you cannot specify an IP address in the **URL** field on the Basic settings. Available values: • **Disabled** means TLS encryption is not used. This value is selected by default. • **Enabled** means TLS encryption is used, but certificates are not verified. • **With verification** means TLS encryption is used with verification of the certificate signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during application installation and are stored on the KUMA Core server in the /opt/kaspersky/kuma/core/certificates/ directory. Obsolete encryption mode, deprecated. Will be removed in future versions of KUMA. • **Custom PFX** means TLS encryption with a PFX secret. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the KUMA web interface as a PFX secret. If you select this value, in the PFX secret drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select Create new. If you want to edit the settings of an existing secret, click the pencil icon next to it. **How to create a PFX secret?** *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and is displayed in the **PFX secret** drop-down list. • **Custom PFX with client authentication** means TLS encryption with a PFX secret. In this mode, the client is authenticated using a certificate. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the KUMA web interface as a PFX secret. If you select this value, in the PFX secret drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select Create new. If you want to edit the settings of an existing secret, click the pencil icon next to it. **How to create a PFX secret?** *To create a PFX secret:* 1. In the **Name** field, enter the name of the PFX secret. 2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key. 3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard. 4. Click the **Create** button. The PFX secret is created and displayed in the **PFX secret** drop-down list.

Sql type

KUMA supports multiple **types of databases**.

The program supports the following types of SQL databases:

- SQLite.
- MariaDB 10.5 or later.
- MSSQL.
- MySQL 5.7 or later.

- PostgreSQL.
- Cockroach.
- Oracle.
- Firebird.

When creating a connector, you must specify general connector settings and specific database connection settings.

On the **Basic settings** tab, you must specify the following settings for the connector:

- **Name** (required)—unique name of the resource. Must contain 1 to 128 Unicode characters.
- **Type** (required)—connector type, **sql**.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Default query** (required)—SQL query that is executed when connecting to the database.
- **Reconnect to the database every time a query is sent** — the check box is cleared by default.
- **Poll interval, ms** —interval for executing SQL queries. This value is specified in seconds. The default value is 10 seconds.
- **Description**—resource description: up to 4,000 Unicode characters.

To connect to the database, you must define the values of the following settings on the **Basic settings** tab, in the **Connection** section:

- In the **Database type** drop-down list, you can select the type of the database you want to connect to. After selecting the type of database, the prefix corresponding to the communication protocol is displayed in the **URL** field. For example, for a ClickHouse database, the **URL** field contains the clickhouse:// prefix.
- If the **Secret separately** check box is selected, the window displays the required **URL** field in which you can specify the connection URL, and a **Secret** drop-down list with secrets of the 'credentials' type. In this way, you can view connection information without having to re-create a large number of connections if the password of the user account that you used for the connections changes. If this check box is cleared, only the **URL** field is available for selecting or creating a secret of the 'urls' type. This check box is cleared by default.
- **URL** (required)—secret that stores a list of URLs for connecting to the database.
  - Field for selecting or creating a secret of the 'urls' type, which stores a list of URLs for connecting to the database. Available if the Secret separately check box is cleared.

    If necessary, you can **edit** or **create a secret**.

    To create a secret:

    1. Click the + button.

       The secret window is displayed.

2. Define the values for the following settings:

   a. **Name**—the name of the added secret.

   b. **Type**. This value is set by default and cannot be changed.

   c. **URL**—URL of the database.

   You must keep in mind that each type of database uses its own URL format for connections. Available URL formats are as follows:

   - For SQLite:

     `sqlite3://file:<file_path>`

     > A question mark (?) is used as a placeholder.

   - For MSSQL:

     - `sqlserver://<user>:<password>@<server:port>/<instance_name>?database=<database>` (recommended)
     - `sqlserver://<user>:<password>@<server>?database=<database>`

     > The characters @p1 are used as a placeholder.

   - For MySQL:

     `mysql://<user>:<password>@tcp(<server>:<port>)/<database>`

     > The characters %s are used as a placeholder.

   - For PostgreSQL:

     `postgres://<user>:<password>@<server>/<database>?sslmode=disable`

     > The characters $1 are used as a placeholder.

   - For Cockroach:

     - `postgres://<user>:<password>@<server>:<port>/<database>?sslmode=disable`

     > The characters $1 are used as a placeholder.

   - For Firebird:

     `firebirdsql://<user>:<password>@<server>:<port>/<database>`

A question mark (?) is used as a placeholder.

d. **Description**—additional information about the secret.

3. If necessary, click **Add** and specify an additional URL. In this case, if one URL is not available, the program connects to the next URL specified in the list of addresses.

4. Click the **Save** button.

To edit the secret:

1. Click the button.

   The secret window is displayed.

2. Specify the values for the settings that you want to change.

   You can change the following values:

   a. **Name**—the name of the added secret.

   b. **URL**—URL of the database.

   You must keep in mind that each type of database uses its own URL format for connections.

   Available URL formats are as follows:

   - For SQLite:

     `sqlite3://file:<file_path>`

     A question mark (?) is used as a placeholder.

   - For MSSQL:

     - `sqlserver://<user>:<password>@<server:port>/<instance_name>?database=<database>` (recommended)

     - `sqlserver://<user>:<password>@<server>?database=<database>`

     The characters @p1 are used as a placeholder.

   - For MySQL:

     `mysql://<user>:<password>@tcp(<server>:<port>)/<database>`

     The characters `?` are used as placeholders.

   - For PostgreSQL:

     `postgres://<user>:<password>@<server>/<database>?sslmode=disable`

The characters $1 are used as a placeholder.

- For Cockroach:

  ```
  postgres://<user>:<password>@<server>:<port>/<database>?sslmode=disable
  ```

  The characters $1 are used as a placeholder.

- For Firebird:

  ```
  firebirdsql://<user>:<password>@<server>:<port>/<database>
  ```

  A question mark (?) is used as a placeholder.

c. **Description**—additional information about the secret.

3. If necessary, click **Add** and specify an additional URL. In this case, if one URL is not available, the program connects to the next URL specified in the list of URLs.

4. Click the **Save** button.

When creating connections, strings containing account credentials with special characters may be incorrectly processed. If an error occurs when creating a connection but you are sure that the settings are correct, enter the special characters in percent encoding.

Codes of special characters

!	#	$	%	&	'	(	)	*	+
%21	%23	%24	%25	%26	%27	%28	%29	%2A	%2B
,	/	:	;	=	?	@	[	]	\
%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D	%5C

The following special characters are not supported in passwords used to access SQL databases: space, [,], :, /, #, %, \.

- Field for specifying the URL of the connection. It is used together with a secret of the 'credentials' type. Available if the **Secret separately** check box is selected.

- **Secret** is a drop-down list for selecting an existing secret or creating a new secret of the 'credentials' type. The drop-down list is available if the **Secret separately** check box is selected.

- **Authorization** is the type of authorization when connecting to the specified URL Possible values:
  - **Disabled** is the default value.
  - If **Plain** is selected, you must specify the secret containing user account credentials for authorization when connecting to the connector.
  - If **PublicPKI** is selected, you must specify the secret containing the base64-encoded PEM private key and the public key.
- **TLS** mode specifies whether TLS encryption is used. Available values:
  - 'Disabled' means TLS encryption is not used.
  - 'Enabled' means encryption is used, but without certificate verification.
  - 'Custom CA' means encryption is used with verification of the certificate that must be signed by a Certificate Authority. The secret containing the certificate is selected from the 'Custom CA' drop-down list, which is displayed when this option is selected.

    Creating a certificate signed by a Certificate Authority

    *To use this **TLS** mode, you must do the following on the KUMA Core server (OpenSSL commands are used in the examples below):*

    1. Create the key that will be used by the Certificate Authority.

       Example command:

       ```
       openssl genrsa -out ca.key 2048
       ```

    2. Generate a certificate for the key that was just created.

       Example command:

       ```
       openssl req -new -x509 -days 365 -key ca.key -subj "/CN=<common host name of
       Certificate Authority>" -out ca.crt
       ```

    3. Create a private key and a request to have it signed by the Certificate Authority.

       Example command:

       ```
       openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=<common host
       name of KUMA server>" -out server.csr
       ```

    4. Create a certificate signed by the Certificate Authority. The subjectAltName must include the domain names or IP addresses of the server for which the certificate is being created.

       Example command:

       ```
       openssl x509 -req -extfile <(printf
       "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in
       server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
       ```

    5. The obtained server.crt certificate should be uploaded in the KUMA Console as a **certificate**-type secret, which should then be selected from the **Custom CA** drop-down list.

    When using TLS, it is impossible to specify an IP address as a URL.

- **Identity column** (required)—name of the column that contains the ID for each row of the table.

- **Identity seed** (required)—identity column value that will be used to determine the specific line to start reading data from the SQL table.

- **Query**—field for an additional SQL query. The query indicated in this field is performed instead of the default query.

- **Poll interval, ms** —interval for executing SQL queries. The interval defined in this field replaces the default interval for the connector.

  This value is specified in seconds. The default value is 10 seconds.

On the **Advanced settings** tab, you need to specify the following settings for the connector:

- **Character encoding**—the specific encoding of the characters. The default value is `UTF-8`.

  > KUMA converts SQL responses to UTF-8 encoding. You can configure the SQL server to send responses in UTF-8 encoding or change the encoding of incoming messages on the KUMA side.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Within a single connector, you can **create a connection** for multiple supported databases.

To create a connection for multiple SQL databases:

1. Click the **Add connection** button.

2. Specify the **URL**, **Identity column**, **Identity seed**, **Query**, and **Poll interval, sec** values.

3. Repeat steps 1–2 for each required connection.

If a collector with a connector of the sql type cannot be started, check if the /opt/kaspersky/kuma/collector/<`collector ID`>/sql/state-<`file ID`> state file is empty.

If that state file is empty, delete it and restart the collector.

Supported SQL types and their specific usage features

The following SQL types are supported:

- MSSQL

  Example URLs:

  - `sqlserver://{user}:{password}@{server:port}/{instance_name}?database={database}` – (recommended option)

  - `sqlserver://{user}:{password}@{server}?database={database}`

  The characters `@p1` are used as a placeholder in the SQL query.

If you need to connect using domain account credentials, specify the account name in `<domain>%5C<user>` format. For example: `sqlserver://domain%5Cuser:password@ksc.example.com:1433/SQLEXPRESS?database=KAV`.

- MySQL

  Example URL: `mysql://{user}:{password}@tcp({server}:{port})/{database}`

  The characters `?` are used as placeholders in the SQL query.

- PostgreSQL

  Example URL: `postgres://{user}:{password}@{server}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- CockroachDB

  Example URL: `postgres://{user}:{password}@{server}:{port}/{database}?sslmode=disable`

  The characters `$1` are used as a placeholder in the SQL query.

- SQLite3

  Example URL: `sqlite3://file:{file_path}`

  A question mark (`?`) is used as a placeholder in the SQL query.

  When querying SQLite3, if the initial value of the ID is in datetime format, you must add a date conversion with the sqlite datetime function to the SQL query. For example: select * from connections where datetime(login_time) > datetime(?, 'utc') order by login_time. In this example, `connections` is the SQLite table, and the value of the variable `?` is taken from the **Identity seed** field, and it must be specified in the {date}T{time}Z format (for example, 2021-01-01T00:10:00Z).

- Oracle DB

  In version 2.1.3 or later, KUMA uses a new driver for connecting to oracle. When upgrading, KUMA renames the connection secret to 'oracle-deprecated' and the connector continues to work. If no events are received after starting the collector with the 'oracle-deprecated' driver type, create a new secret with the 'oracle' driver and use it for connecting.

  We recommend using the new driver.

  Example URL of a secret with the new 'oracle' driver:

  oracle://{user}:{password}@{server}:{port}/{service_name}

  oracle://{user}:{password}@{server}:{port}/?SID={SID_VALUE}

  Example URL of a secret with the legacy 'oracle-deprecated' driver:

  oracle-deprecated://{user}/{password}@{server}:{port}/{service_name}

  The `:val` SQL variable is used as a placeholder in.

When accessing Oracle DB, if the initial ID value is used in the datetime format, you must consider the type of the field in the database itself and, if necessary, add conversions of the time string in the query to ensure correct operation of the sql connector. For example, if the Connections table in the database has a login_time field, the following conversions are possible:

- If the login_time field has the TIMESTAMP type, then depending on the database settings, the login_time field may contain a value in the YYYY-MM-DD HH24:MI:SS format (for example, 2021-01-01 00:00:00). Then, in the **Identity seed** field, specify 2021-01-01T00:00:00Z, and perform the conversion in the query using the to_timestamp function. For example:

  ```
  select * from connections where login_time > to_timestamp(:val, 'YYYY-MM-DD"T"HH24:MI:SS"Z"')
  ```

- If the login_time field has the TIMESTAMP type, then depending on the database settings, the login_time field may contain a value in the YYYY-MM-DD"T"HH24:MI:SSTZH:TZM format (for example, 2021-01-01T00:00:00+03:00). Then, in the **Identity seed** field, specify 2021-01-01T00:00:00+03:00, and perform the conversion in the query using the to_timestamp_tz function. For example:

  ```
  select * from connections_tz where login_time > to_timestamp_tz(:val, 'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM')
  ```

  For more details about the to_timestamp and to_timestamp_tz functions, refer to the official Oracle documentation.

To interact with Oracle DB, you must install the libaio1 Astra Linux package.

- Firebird® SQL

  Example URL:

  ```
  firebirdsql://{user}:{password}@{server}:{port}/{database}
  ```

  A question mark (`?`) is used as a placeholder in the SQL query.

  If a problem occurs when connecting Firebird on Windows, use the full path to the database file. For example:

  ```
  firebirdsql://{user}:{password}@{server}:{port}/C:\Users\user\firebird\db.FDB
  ```

- ClickHouse

  This connector works with ClickHouse only on TCP port 9000 by default without TLS encryption and on port 9440 by default in TLS mode. If the TLS encryption mode is configured on the ClickHouse server, and the 'Disabled' mode is selected in the connector, or vice versa, the database connection is not established.

  If you want to connect to the KUMA ClickHouse, in the SQL connector settings, specify the PublicPki secret type, which contains the base64-encoded PEM private key and the public key.

  In the SQL connector settings for the ClickHouse connection, you must specify a **TLS mode**: the **Disabled** mode is not allowed if a certificate is used for authentication. If you select **Custom CA**, in the **Identity column** field, specify a secret ID of the 'certificate' type.

  You must also specify an **Authorization** type:

  - If **Disabled** is specified, the **Identity column** setting is left unset.
  - **Plain** is used when the **Secret separately** check box is selected and the ID of a secret of the 'credentials' type is specified in the **Identity column** field.
  - **PublicPki** is used when the **Secret separately** check box is selected and the ID of a secret of the 'PublicPki' type is specified in the **Identity column** field.

  The **Secret separately** check box lets you specify the URL separately, not as part of the secret.

A sequential request for database information is supported in SQL queries. For example, if you type `select * from <name of data table> where id > <placeholder>` in the **Query** field, the **Identity seed** field value will be used as the placeholder value the first time you query the table. In addition, the service that utilizes the SQL connector saves the ID of the last read entry, and the ID of this entry will be used as the placeholder value in the next query to the database.

Examples of SQL requests

```
SQLite, Firebird—select * from table_name where id > ?

MSSQL—select * from table_name where id > @p1

MySQL, MariaDB—select * from table_name where id > ?

PostgreSQL, Cockroach—select * from table_name where id > $1

Oracle—select * from table_name where id > :val
```

File type

The **file** type is used to retrieve data from any text file. One string in a file is considered to be one event. Strings delimiter: \n. This type of connector is available for Linux agents and for Windows agents.

To read Windows files, you need to create a connector of the 'file' type and manually install the agent on Windows. In one Windows Agent, you can configure multiple connections of different types, but there must be only one of the 'file' type. The Windows agent must not read its files in the folder where the agent is installed. The connector will work even with a FAT file system; if the disk is defragmented, the connector re-reads all files from scratch because all inodes of files are reset.

We do not recommend running the agent under an administrator account; read permissions for folders/files must be configured for the user account of the agent. We do not recommend installing the agent on important systems; it is preferable to send the logs and read them on dedicated hosts with the agent.

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **file**.
  - **File path** (required)—full path to the file that you need to interact with. For example, `/var/log/*som?[1-9].log or c:\folder\logs.*.` The following paths are not allowed:
    - `` `(?i)^[a-zA-Z]:\\Program Files` ``
    - `` `(?i)^[a-zA-Z]:\\Program Files \(x86\)` ``
    - `` `(?i)^[a-zA-Z]:\\Windows` ``
    - `` `(?i)^[a-zA-Z]:\\ProgramData\\Kaspersky Lab\\KUMA` ``

File and folder mask templates

Masks:

- `'*'` —matches any sequence of characters.
- `'[' [ '^' ] { <range of characters> } ']'` —class of characters (may not be left blank).
- `'?'` —matches any single character.

Ranges of characters:

- `[0-9]` for numerals
- `[a-zA-Z]` for Latin alphabet characters

Examples:

- `/var/log/*som?[1-9].log`
- `/mnt/dns_logs/*/dns.log`
- `/mnt/proxy/access*.log`

Limitations when using prefixes in file paths

Prefixes that cannot be used when specifying paths to files:

- `/*`
- `/bin`
- `/boot`
- `/dev`
- `/etc`
- `/home`
- `/lib`
- `/lib64`
- `/proc`
- `/root`
- `/run`
- `/sys`
- `/tmp`

- `/usr/*`
- `/usr/bin/`
- `/usr/local/*`
- `/usr/local/sbin/`
- `/usr/local/bin/`
- `/usr/sbin/`
- `/usr/lib/`
- `/usr/lib64/`
- `/var/*`
- `/var/lib/`
- `/var/run/`
- `/opt/kaspersky/kuma/`

Files are available at the following paths:

- `/opt/kaspersky/kuma/clickhouse/logs/`
- `/opt/kaspersky/kuma/mongodb/log/`
- `/opt/kaspersky/kuma/victoria-metrics/log/`

Limiting the number of files for watching by mask

The number of files simultaneously watched by mask can be limited by the `max_user_watches` setting of the Core. To view the value of this setting, run the command:

```
cat /proc/sys/fs/inotify/max_user_watches
```

If the number of files for watching exceeds the value of the `max_user_watches` setting, the collector cannot read any more events from the files and the following error is written to the collector log:

```
Failed to add files for watching {"error": "no space left on device"}
```

To make sure that the collector continues to work correctly, you can configure the appropriate rotation of files so that the number of files does not exceed the value of the `max_user_watches` setting, or increase the `max_user_watches` value.

To increase the value of this setting, run the command:

```
sysctl fs.inotify.max_user_watches=< number of files >

sysctl -p
```

> You can also add the value of the `max_user_watches` setting to sysctl.conf so make sure it is kept indefinitely.
>
> After you increase the value of the `max_user_watches` setting, the collector resumes correct operation.

- **Auditd** is the toggle switch of the mechanism that groups auditd log event records received from a connector into a single event. Auditd only supports the \n delimiter, so if the toggle switch is enabled, the **Delimiter** field becomes unavailable. If the **Auditd** toggle switch is enabled in the agent connector, the \n delimiter must be configured in the connector of the collector to which the agent sends events.

- **For Windows** is a toggle switch that, when turned on, enables the receipt of Windows event log events from the Windows agent. In that case, the **Auditd** switch must be turned off. By default, the **For Windows** toggle switch is turned off.

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:

  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

  - **Buffer size** is the setting that lets you specify the size in bytes of the buffer for accumulating events in RAM before sending them for storage or for further processing.

    Default value: 1048576 bytes (1 MB).

    Possible values: positive integer less than or equal to 67108864 bytes (64 MB).

  - **Number of handlers** is the setting that is used to set the number of services processing the queue. You can determine the number of handlers use the formula: (<number of CPU cores>/2) + 2.

  - **Poll interval, ms** is the setting that lets you set the interval with which the connector re-reads the directory with files. The value is in milliseconds. The connector wait for specified time only if there are no changes in the file. If the file is continuously modified, and Poll interval = 5000 milliseconds, the 5-second interval for re-reading the files in the directory is not observed, and instead they are re-read continuously. If there are no changes in the file, the connector waits for 5 seconds. If 0 is set in the web interface, the default value of 700 ms is used. We recommend setting **Poll interval, ms** to at least the **Event buffer TTL** value because otherwise the **Auditd** option may be adversely affected.

  - **Character encoding** setting specifies character encoding. The default value is UTF-8.

  - **Event buffer TTL** is the time to live of the buffer for grouping records into a single auditd event. This field is available if the **Auditd** toggle switch is enabled. The countdown starts the moment when the first event line is received, or immediately after the previous TTL expires. Possible values: 700 ms to 3000 ms. The default value is 2000 ms.

  - **Transport header**—for auditd events, you must specify a regular expression to be used for identifying parts of the auditd log. You can use the default or edit it to suit your needs. The regular expression must contain the record_type_name, record_type_value, and event_sequence_number groups. If a multi-line auditd event contains a prefix, the prefix is retained for the first record, and for subsequent entries, the prefix is discarded.

    You can revert to the original value by clicking **Reset to default value**.

Type 1c-xml

The **1c-xml** type is used to retrieve data from 1C application registration logs. When the connector handles multi-line events, it converts them into single-line events. This type of connector is available for Linux Agents.

When creating this type of connector, specify values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **1c-xml**.
  - **URL** (required)—full path to the directory containing files that you need to interact with. For example, `/var/log/1c/logs/`.

    Limitations when using prefixes in file paths

    Prefixes that cannot be used when specifying paths to files:

    - `/*`
    - `/bin`
    - `/boot`
    - `/dev`
    - `/etc`
    - `/home`
    - `/lib`
    - `/lib64`
    - `/proc`
    - `/root`
    - `/run`
    - `/sys`
    - `/tmp`
    - `/usr/*`
    - `/usr/bin/`
    - `/usr/local/*`
    - `/usr/local/sbin/`
    - `/usr/local/bin/`

- `/usr/sbin/`
- `/usr/lib/`
- `/usr/lib64/`
- `/var/*`
- `/var/lib/`
- `/var/run/`
- `/opt/kaspersky/kuma/`

Files are available at the following paths:

- `/opt/kaspersky/kuma/clickhouse/logs/`
- `/opt/kaspersky/kuma/mongodb/log/`
- `/opt/kaspersky/kuma/victoria-metrics/log/`

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Buffer size** is the setting that lets you specify the size in bytes of the buffer for accumulating events in RAM before sending them for storage or for further processing.

    Default value: 1048576 bytes (1 MB).

    Possible values: positive integer less than or equal to 67108864 bytes (64 MB).
  - **Poll interval, ms** is the setting that lets you set the interval with which the connector re-reads the directory with files. The value is in milliseconds. The connector wait for specified time only if there are no changes in the file. If the file is continuously modified, and Poll interval = 5000 milliseconds, the 5-second interval for re-reading the files in the directory is not observed, and instead they are re-read continuously. If there are no changes in the file, the connector waits for 5 seconds. If 0 is set in the web interface, the default value of 700 ms is used.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Connector operation diagram:

1. The files containing 1C logs with the XML extension are searched within the specified directory. Logs are placed in the directory either manually or using an application written in the 1C language, for example, using the `ВыгрузитьЖурналРегистрации()` function. The connector only supports logs received this way. For more information on how to obtain 1C logs, see the official 1C documentation.

2. Files are sorted by the last modification time in ascending order. All the files modified before the last read are discarded.

   Information about processed files is stored in the file /<collector working directory>/1c_xml_connector/state.ini and has the following format: `offset=<number>\ndev=<number>\ninode=<number>`.

3. Events are defined in each unread file.

4. Events from the file are processed one by one. Multi-line events are converted to single-line events.

Connector limitations:

- Installation of a collector with a 1c-xml connector is not supported in a Windows operating system. To set up file transfers of 1C log files for processing by the KUMA collector:

  1. On the Windows server, grant read access over the network to the folder with the 1C log files.

  2. On the Linux server, mount the shared folder with the 1C log files on the Windows server (see the list of supported operating systems).

  3. On the Linux server, install the collector that you want to process 1C log files from the mounted shared folder.

- Files with an incorrect event format are not read. For example, if event tags in the file are in Russian, the collector does not read such events.

  Example of a correct XML file with an event.

```
<?xml version="1.0" encoding="UTF-8"?>
<v8e:EventLog xmlns: v8e="http://v8.1c.ru/eventLog"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <v8e:Event>
    <v8e:Level>Information</v8e:Level>
    <v8e:Date>2022-12-07T01:55:44+03:00</v8e:Date>
    <v8eApplicationName>generator.go</v8e:ApplicationName>
<v8e:ApplicationPresentation>generator.go</v8e:ApplicationPresentation>
    <v8e:Event>Test event type: Count test</v8e:Event>
    <v8e:EventPresentation></v8e:Event Presentation>
    <v8e:User>abcd_1234</v8e:User>
    <v8e:UserName>TestUser</v8e:UserName>
    <v8e:Computer>Test OC</v8e:Computer>
    <v8e:Metadata></v8e:Metadata>
    <v8e:MetadataPresentation></v8e:MetadataPresentation>
    <v8e:Comment></v8e:Comment>
    <v8e:Data>
      <v8e:Name></v8e:Name>
      <v8e:CurrentOSUser></v8e:CurrentOSUser>
    </v8e:Data>
    <v8e:DataPresentation></v8e:DataPresentation>
    <v8e:TransactionStatus>NotApplicable</v8e:TransactionStatus>
    <v8e:TransactionID></v8e:TransactionID>
    <v8e:Connection>0</v8e:Connection>
    <v8e:Session></v8e:Session>
    <v8e:ServerName>kuma-test</v8e:ServerName>
    <v8e:Port>80</v8e:Port>
    <v8e:SyncPort>0</v8e:SyncPort>
  </v8e:Event>
</v8e:EventLog>
```

Example of a processed event.


```
<v8e:Event><v8e:Level>Information</v8e:Level><v8e:Date>2022-12-
07T01:55:44+03:00</v8e:Date><v8e:ApplicationName>generator.go</v8e:ApplicationName><v8e:ApplicationPresentation>generator.go
</v8e:ApplicationPresentation><v8e:Event>Test event type: Count
test</v8e:Event><v8e:EventPresentation></v8e:EventPresentation><v8e:User>abcd_1234</v8e:User><v8e:UserName>TestUser</v8e:Use
rName><v8e:Computer>Test
OC</v8e:Computer><v8e:Metadata></v8e:Metadata><v8e:MetadataPresentation></v8e:MetadataPresentation><v8e:Comment></v8e:Commen
t><v8e:Data><v8e:Name></v8e:Name><v8e:CurrentOSUser></v8e:CurrentOSUser></v8e:Data><v8e:DataPresentation></v8e:DataPresentat
ion><v8e:TransactionStatus>NotApplicable</v8e:TransactionStatus><v8e:TransactionID></v8e:TransactionID><v8e:Connection>0</v8
e:Connection><v8e:Session></v8e:Session><v8e:ServerName>kuma-
test</v8e:ServerName><v8e:Port>80</v8e:Port><v8e:SyncPort>0</v8e:SyncPort></v8e:Event>
```


- If a file read by the connector is enriched with the new events and if this file is not the last file read in the directory, all events from the file are processed again.

Type 1c-log

The **1c-log** type is used to retrieve data from 1C application technology logs. Strings delimiter: \n. The connector accepts only the first line from a multi-line event record. This type of connector is available for Linux Agents.

When creating this type of connector, specify values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **1c-log**.
  - **URL** (required)—full path to the directory containing files that you need to interact with. For example, `/var/log/1c/logs/`.

    Limitations when using prefixes in file paths

    Prefixes that cannot be used when specifying paths to files:

    - `/*`
    - `/bin`
    - `/boot`
    - `/dev`
    - `/etc`
    - `/home`
    - `/lib`
    - `/lib64`
    - `/proc`
    - `/root`

- `/run`
- `/sys`
- `/tmp`
- `/usr/*`
- `/usr/bin/`
- `/usr/local/*`
- `/usr/local/sbin/`
- `/usr/local/bin/`
- `/usr/sbin/`
- `/usr/lib/`
- `/usr/lib64/`
- `/var/*`
- `/var/lib/`
- `/var/run/`
- `/opt/kaspersky/kuma/`

Files are available at the following paths:

- `/opt/kaspersky/kuma/clickhouse/logs/`
- `/opt/kaspersky/kuma/mongodb/log/`
- `/opt/kaspersky/kuma/victoria-metrics/log/`

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:

  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

  - **Buffer size** is the setting that lets you specify the size in bytes of the buffer for accumulating events in RAM before sending them for storage or for further processing.

    Default value: 1048576 bytes (1 MB).

    Possible values: positive integer less than or equal to 67108864 bytes (64 MB).

- **Poll interval, ms** is the setting that lets you set the interval with which the connector re-reads the directory with files. The value is in milliseconds. The connector wait for specified time only if there are no changes in the file. If the file is continuously modified, and Poll interval = 5000 milliseconds, the 5-second interval for re-reading the files in the directory is not observed, and instead they are re-read continuously. If there are no changes in the file, the connector waits for 5 seconds. If 0 is set in the web interface, the default value of 700 ms is used.

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Connector operation diagram:

1. All 1C technology log files are searched.

   Log file requirements:

   - Files with the LOG extension are created in the log directory (`/var/log/1c/logs/` by default) within a subdirectory for each process.

     Example of a supported 1C technology log structure

```
/var/log/1c
└── logs
    ├── ragent_1426
    │   ├── 22121505.log
    │   ├── 22121506.log
    │   ├── 22121507.log
    │   ├── 22121508.log
    │   ├── 22121509.log
    │   ├── 22121510.log
    │   ├── 22121511.log
    │   ├── 22121512.log
    │   ├── 22121513.log
    │   ├── 22121514.log
    │   ├── 22121515.log
    │   ├── 22121516.log
    │   ├── 22121517.log
    │   └── 22121518.log
    ├── ragent_1427
    │   ├── 22121518.log
    │   ├── 22121519.log
    │   ├── 22121520.log
    │   ├── 22121521.log
    │   ├── 22121522.log
    │   ├── 22121523.log
    │   ├── 22121600.log
    │   ├── 22121601.log
    │   ├── 22121602.log
    │   ├── 22121603.log
    │   ├── 22121604.log
    │   └── 22121605.log
    ├── rmngr_1542
    │   ├── 22121505.log
    │   ├── 22121506.log
    │   ├── 22121507.log
    │   ├── 22121508.log
    │   ├── 22121509.log
    │   ├── 22121510.log
    │   ├── 22121511.log
    │   ├── 22121512.log
    │   ├── 22121513.log
    │   ├── 22121514.log
    │   ├── 22121515.log
    │   ├── 22121516.log
    │   ├── 22121517.log
    │   └── 22121518.log
    ├── rmngr_1544
    │   ├── 22121518.log
    │   ├── 22121519.log
```

   - Events are logged to a file for an hour; after that, the next log file is created.

- The file names have the following format: `<YY><MM><DD><HH>.log`. For example, `22111418.log` is a file created in 2022, in the 11th month, on the 14th at 18:00.

- Each event starts with the event time in the following format: <mm>:<ss>.<microseconds>-<duration_in_microseconds>.

2. The processed files are discarded.

   Information about processed files is stored in the file /<collector working directory>/1c_log_connector/state.json.

3. Processing of the new events starts, and the event time is converted to the RFC3339 format.

4. The next file in the queue is processed.

Connector limitations:

- Installation of a collector with a 1c-log connector is not supported in a Windows operating system. To set up file transfers of 1C log files for processing by the KUMA collector:

  1. On the Windows server, grant read access over the network to the folder with the 1C log files.

  2. On the Linux server, mount the shared folder with the 1C log files on the Windows server (see the list of supported operating systems).

  3. On the Linux server, install the collector that you want to process 1C log files from the mounted shared folder.

- Only the first line from a multi-line event record is processed.

- The normalizer processes only the following types of events:

  - ADMIN

  - ATTN

  - CALL

  - CLSTR

  - CONN

  - DBMSSQL

  - DBMSSQLCONN

  - DBV8DBENG

  - EXCP

  - EXCPCNTX

  - HASP

  - LEAKS

- LIC
- MEM
- PROC
- SCALL
- SCOM
- SDBL
- SESN
- SINTEG
- SRVC
- TLOCK
- TTIMEOUT
- VRSREQUEST
- VRSRESPONSE

Diode type

Used to transmit events using a data diode.

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **diode**.
  - **Data diode destination directory** (required)—full path to the KUMA collector server directory where the data diode moves files containing events from the isolated network segment. After the connector has read these files, the files are deleted from the directory. The path can contain up to 255 Unicode characters.

    Limitations when using prefixes in paths

    Prefixes that cannot be used when specifying paths to files:

    - `/*`
    - `/bin`
    - `/boot`
    - `/dev`

- `/etc`
- `/home`
- `/lib`
- `/lib64`
- `/proc`
- `/root`
- `/run`
- `/sys`
- `/tmp`
- `/usr/*`
- `/usr/bin/`
- `/usr/local/*`
- `/usr/local/sbin/`
- `/usr/local/bin/`
- `/usr/sbin/`
- `/usr/lib/`
- `/usr/lib64/`
- `/var/*`
- `/var/lib/`
- `/var/run/`
- `/opt/kaspersky/kuma/`

Files are available at the following paths:

- `/opt/kaspersky/kuma/clickhouse/logs/`
- `/opt/kaspersky/kuma/mongodb/log/`
- `/opt/kaspersky/kuma/victoria-metrics/log/`

- **Delimiter** is used to specify a character representing the delimiter between events. Available values: `\n`, `\t`, `\0`. If no separator is specified (an empty value is selected), the default value is `\n`.

  This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:

  - **Workers**—the number of services processing the request queue. By default, this value is equal to the number of vCPUs of the KUMA Core server.

  - **Poll interval, sec** —frequency at which the files are read from the directory containing events from the data diode. The default value is 2. The value is specified in seconds.

  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

  - **Compression**—you can use Snappy compression. By default, compression is **disabled**.

    This setting must match for the connector and destination resources used to relay events from an isolated network segment via the data diode.

  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Ftp type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:

  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.

  - **Tenant** (required)—name of the tenant that owns the resource.

  - **Type** (required)—connector type, **ftp**.

  - **URL** (required)—actual URL of the file or file mask beginning with 'ftp://'. For a file mask, you can use * ? [...].

    File mask templates

    Masks:

    - `'*'` —matches any sequence of characters.

    - `'[' [ '^' ] { <range of characters> } ']'` —class of characters (may not be left blank).

    - `'?'` —matches any single character.

    Ranges of characters:

    - `[0-9]` for numerals

    - `[a-zA-Z]` for Latin alphabet characters

Examples:

- `/var/log/*som?[1-9].log`
- `/mnt/dns_logs/*/dns.log`
- `/mnt/proxy/access*.log`

If the URL does not include the FTP server port, port 21 is inserted.

- **URL credentials**—for specifying the user name and password for the FTP server. If there is no user name and password, the line remains empty.
- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Nfs type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **nfs**.
- **URL** (required)—path to the remote folder in the format nfs://host/path.
- **File name mask** (required)—mask used to filter files containing events. Use of masks is acceptable "`*`", "`?`", "`[...]`".
- **Poll interval, sec**—polling interval. The time interval after which files are re-read from the remote system. The value is specified in seconds. The default value is 0.
- **Description**—resource description: up to 4,000 Unicode characters.

Advanced settings tab:

- **Character encoding** setting specifies character encoding. The default value is `UTF-8`.
- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

vmware type

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—connector type, **vmware**.
- **URL** (required)—the URL at which the VMware API is available. The URL must include the host and the port. Only one URL can be specified.
- **VMware credentials** (required) — a secret that stores the username and password for connecting to the VMware API.
- **Client timeout** is the time to wait after a request that did not return events before making a new request. This value is specified in seconds. The default value is 5 seconds. If value is 0, the default value is used.
- **Maximum number of events** — number of events requested from the VMware API in one request. The default value is 100. The maximum value is 1000.
- **Start timestamp**—starting date and time from which you want to read events from the VMware API. The default value is the time when the collector was started. If started after the collector is stopped, the events are read from the last saved date.

Advanced settings tab:

- **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
- **Character encoding**—specifies the character encoding. The default is UTF-8.
- **TLS mode**—TLS encryption mode using certificates in PEM x509 format:
  - **Disabled** (default)—do not use TLS encryption.
  - **Enabled** means encryption is used, but certificates are not verified.
  - **Custom CA**—this option requires you to add a secret with a certificate to the collector. Not self-signed certificate. The certificate of the server must be signed with the certificate specified in the collector settings.
- **Custom CA** (required if **Custom CA** is selected for the TLS mode setting) is the secret where the certificate will be stored.

Wmi type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **wmi**.
  - **URL** (required)—URL of the collector being created, for example: `kuma-collector.example.com:7221`.

    The creation of a collector for receiving data using Windows Management Instrumentation results in the automatic creation of an agent that receives the necessary data on the remote device and forwards that data to the collector service. In the **URL**, you must specify the address of this collector. The URL is known in advance if you already know on which server you plan to install the service. However, this field can also be filled after the Installation Wizard is finished by copying the URL data from the **Resources** → **Active services** section.
  - **Description**—resource description: up to 4,000 Unicode characters.
  - **Default credentials**—drop-down list that does not require any value to be selected. The account credentials used to connect to hosts must be provided in the **Remote hosts** table (see below).
  - The **Remote hosts** table lists the remote Windows assets that you can connect to. Available columns:
    - **Host** (required) is the IP address or name of the device from which you want to receive data. For example, `machine-1`.
    - **Domain** (required)—name of the domain in which the remote device resides. For example, `example.com`.
    - **Log type**—drop-down list to select the name of the Windows logs that you need to retrieve. By default, only preconfigured logs are displayed in the list, but you can add custom logs to the list by typing their name in the **Windows logs** field and then pressing **ENTER**. KUMA service and resource configurations may require additional changes in order to process custom logs correctly.

      Logs that are available by default:
      - Application
      - ForwardedEvents
      - Security
      - System
      - HardwareEvents

If a WMI connection uses at least one log with an incorrect name, the agent that uses the connector does not receive events from all the logs within this connection, even if the names of other logs are specified correctly. The WMI agent connections for which all log names are specified correctly will work properly.

- **Secret**—account credentials for accessing a remote Windows asset with permissions to read the logs. If you leave this field blank, the credentials from the secret selected in the **Default credentials** drop-down list are used. The login in the secret must be specified without the domain. The domain value for access to the host is taken from the **Domain** column of the **Remote hosts** table.

  You can select the secret resource from the drop-down list or create one by clicking the plus icon (+). The selected secret can be changed by clicking the pencil icon (✎).

- **Advanced settings** tab:

  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Receiving events from a remote device

Conditions for receiving events from a remote Windows device hosting a KUMA agent:

- To start the KUMA agent on the remote device, you must use an account with the **Log on as a service** permissions.

- To receive events from the KUMA agent, you must use an account with Event Log Readers permissions. For domain servers, one such user account can be created so that a group policy can be used to distribute its rights to read logs to all servers and workstations in the domain.

- TCP ports 135, 445, and 49152–65535 must be opened on the remote Windows devices.

- You must run the following services on the remote machines:

  - Remote Procedure Call (RPC)

  - RPC Endpoint Mapper

Wec type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:

  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.

  - **Tenant** (required)—name of the tenant that owns the resource.

  - **Type** (required)—connector type, **wec**.

- **URL** (required)—URL of the collector being created, for example: `kuma-collector.example.com:7221`.

  The creation of a collector for receiving data using Windows Event Collector results in the automatic creation of an agent that receives the necessary data on the remote device and forwards that data to the collector service. In the **URL**, you must specify the address of this collector. The URL is known in advance if you already know on which server you plan to install the service. However, this field can also be filled after the Installation Wizard is finished by copying the URL data from the **Resources** → **Active services** section.

- **Description**—resource description: up to 4,000 Unicode characters.

- **Windows logs** (required)—Select the names of the Windows logs you want to retrieve from this drop-down list. By default, only preconfigured logs are displayed in the list, but you can add custom logs to the list by typing their name in the **Windows logs** field and then pressing **ENTER**. KUMA service and resource configurations may require additional changes in order to process custom logs correctly.

  Preconfigured logs:

  - Application
  - ForwardedEvents
  - Security
  - System
  - HardwareEvents

  > If the name of at least one log is specified incorrectly, the agent using the connector does not receive events from any log, even if the names of other logs are correct.

- **Advanced settings** tab:

  - **Character encoding** setting specifies character encoding. The default value is `UTF-8`.

  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

To start the KUMA agent on the remote device, you must use a service account with the "Log on as a service" permissions. To receive events from the operating system log, the service user account must also have Event Log Readers permissions.

You can create one user account with "Log on as a service" and "Event Log Readers" permissions, and then use a group policy to extend the rights of this account to read the logs to all servers and workstations in the domain.

We recommend that you disable interactive logon for the service account.

snmp type

> To process events received via SNMP, you must use json normalizer.

It is available for Windows and Linux Agents. Supported protocol versions:

- snmpV1
- snmpV2
- snmpV3

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **snmp**.
  - **SNMP version** (required)—This drop-down list allows you to select the version of the protocol to use.
  - **Host** (required)—hostname or its IP address. Available formats: hostname, IPv4, IPv6.
  - **Port** (required)—port for connecting to the host. Typically 161 or 162 are used.

  The **SNMP version**, **Host** and **Port** settings define one connection to a SNMP resource. You can create several such connections in one connector by adding new ones by clicking the **SNMP resource** button. You can delete connections by clicking the bin icon (🗑).

  - **Secret** (required)—a drop-down list to select the secret which stores the credentials for connecting via the Simple Network Management Protocol. The secret type must match the SNMP version. If required, a secret can be created in the connector creation window by clicking the plus icon (+). The selected secret can be changed by clicking the pencil icon (✎).
  - **Source data**—a table in which you can specify the rules for naming the received data, according to which OIDs, object identifiers, will be converted into keys with which the normalizer can interact. Available table columns:
    - **Parameter name** (required)—an arbitrary name for the data type. For example, `Site name` or `Site uptime`.
    - **OID** (required)—a unique identifier that determines where to look for the required data at the event source. For example, `1.3.6.1.2.1.1.5`.
    - **Key** (required)—a unique identifier returned in response to a request to the asset with the value of the requested setting. For example, `sysName`. This key can be accessed when normalizing data.
    - **MAC address**—if this functionality is enabled, KUMA correctly decodes data where the OID contains information about the MAC address in OctetString format. After decoding, the MAC address is converted to a String value of the XX:XX:XX:XX:XX:XX format.
  - **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding**—a setting that specifies character encoding. The default value is `UTF-8`.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

snmp-trap type

The **snmp-trap** connector is used in agents and collectors to passively receive SNMP trap messages. The connector receives and prepares messages for normalization by mapping the SNMP object IDs to the temporary keys. Then the message is passed to the JSON normalizer, where the temporary keys are mapped to the KUMA fields and an event is generated.

> To process events received via SNMP, you must use json normalizer.

It is available for Windows and Linux Agents. Supported protocol versions:

- snmpV1
- snmpV2

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **snmp-trap**.
  - **SNMP version** (required)—in this drop-down list, select the version of the protocol to be used: **snmpV1** or **snmpV2**.

    > For example, Windows uses the **snmpV2** version by default.

  - **URL** (required) – URL where SNMP Trap messages will be expected. Available formats: hostname:port, IPv4:port, IPv6:port, :port.

  The **SNMP version** and **URL** parameters define one connection used to receive SNMP Traps. You can create several such connections in one connector by adding new ones by clicking the **SNMP resource** button. You can delete connections by clicking the ✕ button.

  - **Source data**—a table in which you specify the rules for naming the received data, according to which OIDs (object identifiers) are converted to the keys with which the normalizer can interact.

    You can click **Apply OIDs for WinEventLog** to populate the table with mappings for OID values that arrive in WinEventLog logs. If more data needs to be determined and normalized in the incoming events, add to the table rows containing OID objects and their keys.

Available table columns:

- **Parameter name** —an arbitrary name for the data type. For example, `Site name` or `Site uptime`.
- **OID** (required)—a unique identifier that determines where to look for the required data at the event source. For example, `1.3.6.1.2.1.1.1`.
- **Key** (required)—a unique identifier returned in response to a request to the asset with the value of the requested setting. For example, `sysDescr`. This key can be accessed when normalizing data.
- **MAC address**—if this functionality is enabled, KUMA correctly decodes data where the OID contains information about the MAC address in OctetString format. After decoding, the MAC address is converted to a String value of the XX:XX:XX:XX:XX:XX format.

Data is processed according to the allow list principle: objects that are not specified in the table are not sent to the normalizer for further processing.

- **Description**—resource description: up to 4,000 Unicode characters.

- **Advanced settings** tab:
  - **Character encoding**—a setting that specifies character encoding. The default value is `UTF-8`. When receiving snmp-trap events from Windows with Russian localization, if you encounter invalid characters in the event, we recommend changing the character encoding in the snmp-trap connector to Windows 1251.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

Configuring the source of SNMP trap messages for Windows

Configuring a Windows device to send SNMP trap messages to the KUMA collector involves the following steps:

1. **Configuring and starting the SNMP and SNMP trap services**
2. **Configuring the Event to Trap Translator service**

Events from the source of SNMP trap messages must be received by the KUMA collector, which uses a connector of the snmp-trap type and a json normalizer.

Configuring and starting the SNMP and SNMP trap services

To configure and start the SNMP and SNMP trap services in Windows 10:

1. Open **Settings** → **Apps** → **Apps and features** → **Optional features** → **Add feature** → **Simple Network Management Protocol (SNMP)** and click **Install**.
2. Wait for the installation to complete and restart your computer.
3. Make sure that the SNMP service is running. If any of the following services are not running, enable them:
   - **Services** → **SNMP Service**.
   - **Services** → **SNMP Trap**.

4. Right-click **Services** → **SNMP Service**, and in the context menu select **Properties**. Specify the following settings:

   - On the **Log On** tab, select the **Local System account** check box.

   - On the **Agent** tab, fill in the **Contact** (for example, specify `User-win10`) and Location (for example, specify `detroit`) fields.

   - On the **Traps** tab:

     - In the **Community Name** field, enter **community public** and click **Add to list**.

     - In the **Trap destination** field, click **Add**, specify the IP address or host of the KUMA server on which the collector that waits for SNMP events is deployed, and click **Add**.

   - On the **Security** tab:

     - Select the **Send authentication trap** check box.

     - In the **Accepted community names** table, click **Add**, enter **Community Name public** and specify **READ WRITE** as the **Community rights**.

     - Select the **Accept SNMP packets from any hosts** check box.

5. Click **Apply** and confirm your selection.

6. Right click **Services** → **SNMP Service** and select **Restart**.

To configure and start the SNMP and SNMP trap services in Windows XP:

1. Open **Start** → **Control Panel** → **Add or Remove Programs** → **Add / Remove Windows Components** → **Management and Monitoring Tools** → **Details**.

2. Select **Simple Network Management Protocol** and **WMI SNMP Provider**, and then click **OK** → **Next**.

3. Wait for the installation to complete and restart your computer.

4. Make sure that the SNMP service is running. If any of the following services are not running, enable them by setting the **Startup type** to **Automatic**:

   - **Services** → **SNMP Service**.

   - **Services** → **SNMP Trap**.

5. Right-click **Services** → **SNMP Service**, and in the context menu select **Properties**. Specify the following settings:

   - On the **Log On** tab, select the **Local System account** check box.

   - On the **Agent** tab, fill in the **Contact** (for example, specify `User-win10`) and Location (for example, specify `detroit`) fields.

- On the **Traps** tab:
  - In the **Community Name** field, enter **community public** and click **Add to list**.
  - In the **Trap destination** field, click **Add**, specify the IP address or host of the KUMA server on which the collector that waits for SNMP events is deployed, and click **Add**.
- On the **Security** tab:
  - Select the **Send authentication trap** check box.
  - In the **Accepted community names** table, click **Add**, enter **Community Name public** and specify **READ WRITE** as the **Community rights**.
  - Select the **Accept SNMP packets from any hosts** check box.

6. Click **Apply** and confirm your selection.
7. Right click **Services** → **SNMP Service** and select **Restart**.

Changing the port for the SNMP trap service

You can change the SNMP trap service port if necessary.

To change the port of the SNMP trap service:

1. Open the C:\Windows\System32\drivers\etc folder.
2. Open the **services** file in Notepad as an administrator.
3. In the **service name** section of the file, specify the snmp-trap connector port added to the KUMA collector for the **SNMP trap** service.
4. Save the file.
5. Open the Control Panel and select **Administrative Tools** → **Services**.
6. Right-click **SNMP Service** and select **Restart**.

Configuring the Event to Trap Translator service

To configure the Event to Trap Translator service that translates Windows events to SNMP trap messages:

1. In the command line, type `evntwin` and press **Enter**.
2. Under **Configuration type**, select **Custom**, and click the **Edit** button.
3. In the **Event sources** group of settings, click the **Add** button to find and add the events that you want to send to KUMA collector with the SNMP trap connector installed.
4. Click the **Settings** button, in the opened window, select the **Don't apply throttle** check box, and click **OK**.
5. Click **Apply** and confirm your selection.

elastic type

Support is guaranteed for Elasticsearch version 7.0.0.

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **elastic**.
  - **URL** (required)—valid URL of the Elasticsearch server.
  - **Elastic credentials**—drop-down list in which you can select the secret which stores the credentials for connecting to the Elasticsearch server.
  - **Elastic fingerprint**—drop-down list for selecting a secret that stores secrets of the `fingerprint` type for connecting to the Elasticsearch server and secrets of the `certificate` type for using a CA certificate.
  - **Index** (required)—Name of the index in Elasticsearch.
  - **Query** (required)—query to Elasticsearch. We recommend specifying the `size` parameter in the query to prevent performance issues with KUMA and Elasticsearch.

    Query example:

    ```
    "query" : { "match_all" : {} },"size" : 25
    ```
  - **Sorting** (required)—sorting order. Possible values: asc, desc.
  - **Poll interval, sec**—interval between queries to the Elasticsearch server in seconds if the previous query did not return any events. If Elasticsearch contained events at the time of the request, the connector will receive events until all available events have been received from Elasticsearch.
  - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:
  - **Character encoding** setting specifies character encoding. The default value is UTF-8.
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.

etw type

When creating this type of connector, you need to define values for the following settings:

- **Basic settings** tab:
  - **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
  - **Tenant** (required)—name of the tenant that owns the resource.
  - **Type** (required)—connector type, **etw**.
  - **URL** (required)—valid URL of the DNS server.
  - **Session name** (required)—you can specify only one session name, which corresponds to the Microsoft-Windows-DNSServer ETW provider {EB79061A-A566-4698-9119-3ED2807060E7}.
  - **Extract event information**—if the toggle switch is turned off, the minimum set of event information is extracted that can be obtained without having to download third-party metadata from the disk. This method helps conserve CPU resources on the computer with the agent. By default, all data about the event is extracted.
  - **Extract event properties**—if the toggle switch is turned off, event properties are not extracted, and this helps conserve CPU usage on the computer with the agent. By default, event properties are extracted. You can use the **Extract event properties** toggle switch only when the **Extract event information** toggle switch is on.
  - **Description**—resource description: up to 4,000 Unicode characters.
- **Advanced settings** tab:
  - **Debug**—a toggle switch that lets you specify whether resource logging must be enabled. By default, this toggle switch is in the **Disabled** position.
  - **Character encoding**—used to specify the source encoding in UTF-8. We recommend editing this setting only if garbled characters are displayed in the fields of the normalized event. By default, no value is set.
  - **TLS mode**—TLS encryption mode using certificates in PEM x509 format:
    - **Disabled** (default)—do not use TLS encryption.
    - **Enabled** means encryption is used, but certificates are not verified.
    - **With verification**—use encryption with verification that the certificate was signed with the KUMA root certificate. The root certificate and key of KUMA are created automatically during program installation and are stored on the KUMA Core server in the /opt/kaspersky/kuma/core/certificates/ folder.
  - **Compression**—you can use Snappy compression. By default, compression is disabled.

universalAPI type

Connectors of the **universalAPI** type are used for receiving data from any REST APIs that support JSON responses.

This connector type is designed only for REST APIs that can be configured via the user interface. These connectors always use the **json** normalizer. Only one connection can be created for one connector. Only one URL can be configured for a connection.

The **universalAPI** connector only supports collecting events from JSON structures with one nesting level. If the response structure contains nested lists (for example, a list of objects that each have another list of objects inside), the connector does not support collecting events from such lists. In this case, we recommend collecting top-level data and processing nested structures at the normalizer level.

You can test the connection to the source after running the collector by analyzing its logs. In case of problems with the connection or authorization issues, the collector is displayed in the **Resources and services** → **Active services** section with a yellow status.

> The **universalAPI** connector does not replace other existing API connectors, such as **office365**.

When creating this type of connector, you need to define values for the following settings:

Basic settings tab:

- **Name**—unique name of the resource. The maximum length of the name is 128 Unicode characters. Required setting.
- **Tenant**—the name of the tenant that owns the resource. Required setting.
- **Type**—type of the connector, **universalAPI**. Required setting.
- **Tags** for resource search.
- **Description**—the description of a resource. The maximum length of the description is 4000 Unicode characters.
- **URL** of the REST API server. You do not need to specify the connection protocol (http or https). It is inferred automatically from the tlsMode setting. You can only specify one URL. Required setting.
- **Request interval (seconds)** for the specified URL. Enter a number from 60 to 3600 (that is, from 1 minute to 60 minutes). The default value is 3600.
- **HTTP method** used for API requests. Available values:
  - **GET**
  - **POST**
  - **PUT**
  - **DELETE**

Required setting.

- **Authorization** method for connecting to the API. The following authorization methods are available:

  - **Disabled**. No authorization.

  - **Plain**. Login and password authorization. You need to use a secret of the **credentials** type.

  - **Token**. Token authorization. You need to use a secret of the **token** type.

  For more information, see the section on secrets.

  The default value is **Disabled**.

  If on the **Advanced settings** tab, the **TLS mode** field is set to **Custom PFX**, the **Authorization** field is not available.

- The **Secret ID** field is available if the **Authorization** field is set to **Plain**. You need to select a secret of the **credentials** type. Required setting.

- The **Token** field is available if the **Authorization** field is set to **Token**. You need to select a secret of the **token** type. Required setting.

- The **Token name** field is available if the **Authorization** field is set to **Token**. Available values:

  - **Bearer**

  - **OAuth**

  You can also specify the name manually. Required setting.

- **Query body** is the body of the HTTP request. You can also use variables: `$token`, `$username`, `$password`, `$identityValue`. If iteration is enabled and the state has an iteration key, it is automatically substituted into the request parameters.

- **Data path in JSONPath format** is the path to data in structure of the the API response that points to an object or list of objects. The path is specified in JSONPath format (for example, `$.data`, `$.data.edges[0].node.event`, `$.data.edges[*].node.event`). The `$` character it is automatically substituted into the field and you do not need to enter it. Required setting.

- The **Identity field** toggle switch enables an unique ID by which events are to be selected. The toggle switch is turned off by default.

- The **Field path in JSONPath format** field is available if the **Identity field** check box is selected. The path to the ID field in the structure of the API response element is specified in the JSONPath format (for example, `$.id`). The `$` character is automatically substituted into the field and you do not need to enter it. Required setting.

- The **Type** field is available if the **Identity field** toggle switch is enabled. Type of the ID field by which the events will be selected. Available values:

  - **int**

  - **float**

  - **string**

  - **date**

Internally, the value is stored as a timestamp in milliseconds. Required setting.

- The **Date format** field is available if the **Type** field is set to **date**. The date format can include the date, time, fractions of a second, and time zone, for example, `YYYY-MM-DDTHH:mm:ssX`. The specified format is used:
  - For parsing the initial value of the ID.
  - For parsing ID values from REST API responses. If the date format does include a time zone, the value is processed in the specified time zone. If the date format does not include a time zone, the value is processed as UTC.
  - For generating date strings for query parameters (e.g., `$identityValue` in `queryParams`, `headers`, `body`). If the date format includes a time zone, the string is generated in the server's time zone. If the date format does not include a time zone, the string is generated as UTC.
- The **Initial value** field is available if the **Identity field** toggle switch is enabled. The identity seed is specified as a string in the format corresponding to the selected **Type**. For the **date** type, you need to specify the value in the format specified in the **Date format** field. This value can contain Unicode characters. The minimum length is 1 character.
- The **Iteration parameters** toggle switch enables the iteration mode used for APIs with staged data output. The toggle switch is turned off by default.
- **Iteration key field path in JSONPath format** is the path to the iteration key field in the structure of the API response element in JSONPath format. The `$` character is automatically inserted into the field. Required setting.
- **Request interval in iterative mode (seconds)** is the URL poll interval in iteration mode, in seconds. The default value is `60`.
- **URL parameter for iteration key** is the name of the URL parameter in which the iteration key is passed when making a request to the API.
- **Optional query parameters** are additional request parameters in the `map[string][]string` format. You can also use variables: `$token`, `$username`, `$password`, `$identityValue`. To add a parameter, click the **+ Add** button and fill in the **Key** and **Value** fields. To delete a parameter, select the check box next to it and click **Delete**.
- **Query headers** are optional HTTP request headers in the `map[string][]string` format. You can also use variables: `$token`, `$username`, `$password`, `$identityValue`. To add a parameter, click the **+ Add** button and fill in the **Key** and **Value** fields. To delete a parameter, select the check box next to it and click **Delete**.
- **Additional fields** are fields that are added to each data element in the `map[string][]string` format. To add fields, click the **+ Add** button and fill in the **Field name** and **JSONPath** fields. To delete a field, select the check box next to it and click **Delete**.

Advanced settings tab:

- The **Debug** toggle switch enables debug mode for the connector. Turned off by default.
- **Character encoding**. The default is **UTF-8**.
- **Proxy** specifies proxy server settings, if one is needed to connect to the API. You can select one of the available proxy servers or create a new proxy server.

- **TLS mode**. When using TLS encryption, you cannot specify an IP address in the **URL** field on the **Basic settings** tab. Available values:

  - **Disabled** means TLS encryption is not used. This value is selected by default.

  - **Enabled** means TLS encryption is used, but certificates are not verified.

  - **Custom CA** enables TLS encryption with validation of the certificate signed by a Certificate Authority. If you select this value, in the **Custom CA** drop-down list, specify a secret with a certificate signed by a certification authority. You can select an existing secret or create a new secret. To create a new secret, select **Create new**.

    If you want to edit the settings of an existing secret, click the pencil icon next to it.

    How to create a certificate signed by a Certificate Authority?

    You can create a CA-signed certificate on the KUMA Core server (the following command examples use OpenSSL).

    To create a certificate signed by a Certificate Authority:

    1. Generate a key to be used by the Certificate Authority, for example:

       ```
       openssl genrsa -out ca.key 2048
       ```

    2. Create a certificate for the generated key, for example:

       ```
       openssl req -new -x509 -days 365 -key ca.key -subj "/CN=< common host name of Certificate Authority >" -out ca.crt
       ```

    3. Create a private key and a request to have it signed by the Certificate Authority, for example:

       ```
       openssl req -newkey rsa:2048 -nodes -keyout server.key -subj "/CN=< common host name of KUMA server >" -out server.csr
       ```

    4. Create the certificate signed by the Certificate Authority. You need to include the domain names or IP addresses of the server for which you are creating the certificate in the `subjectAltName` variable, for example:

       ```
       openssl x509 -req -extfile <(printf "subjectAltName=DNS:domain1.ru,DNS:domain2.com,IP:192.168.0.1") -days 365 -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt
       ```

    5. Upload the created `server.crt` certificate in the KUMA web interface to a secret of the **certificate** type, then in the **Custom CA** drop-down list, select the secret of the **certificate** type.

  - **Custom PFX PFX** enables TLS encryption with a PFX secret. You must generate a PFX certificate with a private key in PKCS#12 container format, then upload the PFX certificate to the Kaspersky Security Center Web Console as a PFX secret. If you select this value, in the **Custom PFX** drop-down list, specify a PFX secret with a certificate signed by a certification authority. You can select an existing PFX secret or create a new PFX secret. To create a new PFX secret, select **Create new**.

    If you want to edit the settings of an existing secret, click the pencil icon next to it.

    How to create a PFX secret?

To create a PFX secret:

1. In the **Name** field, enter the name of the PFX secret.
2. Click **Upload PFX** and select the PKCS#12 container file to which you exported the PFX file with the private key.
3. In the **Password** field, enter the password of the PFX file that was set in the PFX Export Wizard.
4. Click the **Create** button.

The PFX secret is created and is displayed in the **PFX secret** drop-down list.

- **Request timeout (seconds)** is the time to wait for a response to an API request, in seconds. The default response timeout is 30 seconds.
- **Request retry count** is the number of repeated attempts to make a request if an error occurs. The default value is `1`.
- **Retry interval (sec.)** is the interval between request retries, in seconds. The default value is `0`.

Predefined connectors

The connectors listed in the table below are included in the OSMP distribution kit.

Predefined connectors

Connector name	Comment
[OOTB] Continent SQL	Obtains events from the database of the Continent hardware and software encryption system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] InfoWatch Trafic Monitor SQL	Obtains events from the database of the InfoWatch Traffic Monitor system. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC MSSQL	Obtains events from the MS SQL database of the Kaspersky Security Center application. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC MySQL	Obtains events from the MySQL database of the Kaspersky Security Center application. To use it, you must configure the settings of the corresponding secret type.
[OOTB] KSC PostgreSQL	Obtains events from the PostgreSQL database of the Kaspersky Security Center 15.0 application. To use it, you must configure the settings of the corresponding secret type.
[OOTB] Oracle Audit Trail SQL	Obtains audit events from the Oracle database. To use it, you must configure the settings of the corresponding secret type.
[OOTB] SecretNet SQL	Obtains events from the SecretNet SQL database. To use it, you must configure the settings of the corresponding secret type.

Secrets

Secrets are used to securely store sensitive information such as user names and passwords that must be used by KUMA to interact with external services. If a secret stores account data such as user login and password, when the collector connects to the event source, the user account specified in the secret may be blocked in accordance with the password policy configured in the event source system.

Secrets can be used in the following KUMA services and features:

- Collector (when using TLS encryption).
- Connector (when using TLS encryption).
- Destinations (when using TLS encryption or authorization).
- Proxy servers.

Available settings:

- **Name** (required)—a unique name for this type of resource. Must contain 1 to 128 Unicode characters.
- **Tenant** (required)—name of the tenant that owns the resource.
- **Type** (required)—the type of secret.

  When you select the type in the drop-down list, the parameters for configuring this secret type also appear. These parameters are described below.
- **Description**—up to 4,000 Unicode characters.

Depending on the secret type, different fields are available. You can select one of the following secret types:

- **credentials**—this type of secret is used to store account credentials required to connect to external services, such as SMTP servers. If you select this type of secret, you must fill in the **User** and **Password** fields. If the **Secret** resource uses the 'credentials' type to connect the collector to an event source, for example, a database management system, the user account specified in the secret may be blocked in accordance with the password policy configured in the event source system.
- **token**—this secret type is used to store tokens for API requests. Tokens are used when connecting to IRP systems, for example. If you select this type of secret, you must fill in the **Token** field.
- **ktl**—this secret type is used to store Kaspersky Threat Intelligence Portal account credentials. If you select this type of secret, you must fill in the following fields:
  - **User** and **Password** (required fields)—user name and password of your Kaspersky Threat Intelligence Portal account.
  - **PFX file** (required)—lets you upload a Kaspersky Threat Intelligence Portal certificate key.
  - **PFX password** (required)—the password for accessing the Kaspersky Threat Intelligence Portal certificate key.
- **urls**—this secret type is used to store URLs for connecting to SQL databases and proxy servers. In the **Description** field, you must provide a description of the connection for which you are using the secret of **urls** type.

  You can specify URLs in the following formats: hostname:port, IPv4:port, IPv6:port, :port.

- **pfx**—this type of secret is used for importing a PFX file containing certificates. If you select this type of secret, you must fill in the following fields:
  - **PFX file** (required)—this is used to upload a PFX file. The file must contain a certificate and key. PFX files may include CA-signed certificates for server certificate verification.
  - **PFX password** (required)—this is used to enter the password for accessing the certificate key.
- **kata/edr**—this type of secret is used to store the certificate file and private key required when connecting to the Kaspersky Endpoint Detection and Response server. If you select this type of secret, you must upload the following files:
  - Certificate file—KUMA server certificate.

    The file must be in PEM format. You can upload only one certificate file.
  - Private key for encrypting the connection—KUMA server RSA key.

    The key must be without a password and with the PRIVATE KEY header. You can upload only one key file.

    > You can generate certificate and key files by clicking the button.
- **snmpV1**—this type of secret is used to store the values of **Community** access (for example, `public` or `private`) that is required for interaction over the Simple Network Management Protocol.
- **snmpV3**—this type of secret is used for storing data required for interaction over the Simple Network Management Protocol. If you select this type of secret, you must fill in the following fields:
  - **User**—user name indicated without a domain.
  - **Security Level**—security level of the user.
    - **NoAuthNoPriv**—messages are forwarded without authentication and without ensuring confidentiality.
    - **AuthNoPriv**—messages are forwarded with authentication but without ensuring confidentiality.
    - **AuthPriv**—messages are forwarded with authentication and ensured confidentiality.

    You may see additional settings depending on the selected level.
  - **Password**—SNMP user authentication password. This field becomes available when the **AuthNoPriv** or **AuthPriv** security level is selected.
  - **Authentication Protocol**—the following protocols are available: MD5, SHA, SHA224, SHA256, SHA384, SHA512. This field becomes available when the **AuthNoPriv** or **AuthPriv** security level is selected.
  - **Privacy Protocol**—protocol used for encrypting messages. Available protocols: DES, AES. This field becomes available when the **AuthPriv** security level is selected.
  - **Privacy password**—encryption password that was set when the SNMP user was created. This field becomes available when the **AuthPriv** security level is selected.
- **certificate**—this secret type is used for storing certificate files. Files are uploaded to a resource by clicking the **Upload certificate file** button. X.509 certificate public keys in Base64 are supported.

Predefined secrets

The secrets listed in the table below are included in the OSMP distribution kit.

Predefined secrets

Secret name	Description
[OOTB] Continent SQL connection	Stores confidential data and settings for connecting to the APKSh Kontinent database. To use it, you must specify the login name and password of the database.
[OOTB] KSC MSSQL connection	Stores confidential data and settings for connecting to the MS SQL database of Kaspersky Security Center (KSC). To use it, you must specify the login name and password of the database.
[OOTB] KSC MySQL Connection	Stores confidential data and settings for connecting to the MySQL database of Kaspersky Security Center (KSC). To use it, you must specify the login name and password of the database.
[OOTB] Oracle Audit Trail SQL Connection	Stores confidential data and settings for connecting to the Oracle database. To use it, you must specify the login name and password of the database.
[OOTB] SecretNet SQL connection	Stores confidential data and settings for connecting to the MS SQL database of the SecretNet system. To use it, you must specify the login name and password of the database.

Context tables

A *context table* is a container for a data array that is used by KUMA correlators for analyzing events in accordance with correlation rules. You can create context tables in the **Monitoring** → **Resources** section. The context table data is stored only in the correlator to which it was added using filters or actions in correlation rules.

You can populate context tables automatically using correlation rules of 'simple' and 'operational' types or import a file with data for the context table.

You can add, copy, and delete context tables, as well as edit their settings.

Context tables can be used in the following KUMA services and features:

- Correlation rules.
- Dashboard.

The same context table can be used in multiple correlators. However, a separate entity of the context table is created for each correlator. Therefore, the contents of the context tables used by different correlators are different even if the context tables have the same name and ID.

Only data based on correlation rules of the correlator are added to the context table.

You can add, edit, delete, import, and export records in the context table of the correlator.

During the correlation process, when entries are deleted from context tables, service events are generated in the correlators. These events only exist in the correlators, and they are not redirected to other destinations. Service events are sent for processing by correlation rules of that correlator which uses the context table. Correlation rules can be configured to track these events so that they can be used to identify threats.

Service event fields for deleting an entry from a context table are described below.

Event field	Value or comment
`ID`	Event ID.
`Timestamp`	Time when the expired entry was deleted.

`Name`	`"context table record expired"`
`DeviceVendor`	`"Kaspersky"`
`DeviceProduct`	`"KUMA"`
`ServiceID`	Correlator ID.
`ServiceName`	Correlator name.
`DeviceExternalID`	Context table ID.
`DevicePayloadID`	Key of the expired entry.
`BaseEventCount`	Number of updates for the deleted entry, incremented by one.
`FileName`	Name of the context table.

`S.<context table field>`
`SA.<context table field>`
`N.<context table field>`
`NA.<context table field>`
`F.<context table field>`
`FA.<context table field>`
	Depending on the type of the entry that dropped out from the context table, the dropped-out context table entry is recorded in the corresponding type of event:
	for example, S.`<context table field>`=`<context table field value>`
	SA.`<context table field>`=`<array of context table field values>`
	
	Context table records of the boolean type have the following format:
	S.`<context table field>`=true/false
	SA.`<context table field>`=false,true,false

Viewing the list of context tables

To view the context table list of the correlator:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. In the context menu of the correlator for which you want to view context tables, select **Go to context tables**.

   The **Correlator context tables** list is displayed.

The table contains the following data:

- **Name**—name of the context table.
- **Size on disk**—size of the context table.
- **Directory**—path to the context table on the KUMA correlator server.

Adding a context table

To add a context table:

1. In the main menu, go to **Monitoring** → **Resources** → **Context tables**.

2. In the **Context tables** window, click **Add**.

   This opens the **Create context table** window.

3. In the **Name** field, enter a name for the context table.

4. In the **Tenant** drop-down list, select the tenant that owns the resource.

5. In the **TTL** field, specify time the record added to the context table is stored in it.

   When the specified time expires, the record is deleted. The time is specified in seconds. The maximum value is `31536000` (1 year).

   The default value is `0`. If the value of the field is 0, the record is stored indefinitely.

6. In the **Description** field, provide any additional information.

   You can use up to 4,000 Unicode characters.

   This field is optional.

7. In the **Schema** section, specify which fields the context table has and the data types of the fields.

   Depending on the data type, a field may or may not be a key field. At least one field in the table must be a key field. The names of all fields must be unique.

   To add a table row, click **Add** and fill in the table fields:

   a. In the **Name** field, enter the name of the field. The maximum length is 128 characters.

   b. In the **Type** drop-down list, select the data type for the field.

   Possible field data types

Possible data types of context table fields

Field data type	Can be a key field	Comment
Integer	Yes	—
Floating point number	Yes	—
String	Yes	—
Boolean	Yes	—
Timestamp	Yes	For a field of this type, it is checked that the field value is greater than or equal to zero. No other operations are provided.
IP address	Yes	For a field of this type, it is checked that the field value corresponds to the IPv4, IPv6 format. No other operations are provided.
Integer list	No	—
Float list	No	—
List of strings	No	—

Field data type	Can be a key field	Comment
Boolean list	No	—
Timestamp list	No	For a field of this type, it is checked that each item in the list is greater than or equal to zero. No other operations are provided.
IP list	No	For a field of this type, it is checked that each item of the list corresponds to the IPv4, IPv6 format. No other operations are provided.

c. If you want to make a field a key field, select the **Key field** check box.

A table can have multiple key fields. Key fields are chosen when the context table is created, uniquely identify a table entry and cannot be changed.

If a context table has multiple key fields, each table entry is uniquely identified by multiple fields (composite key).

8. Add the required number of context table rows.

After saving the context table, the schema cannot be changed.

9. Click the **Save** button.

The context table is added.

Viewing context table settings

To view the context table settings:

1. In the main menu, go to **Monitoring** → **Resources** → **Context tables**.

2. In the list in the **Context tables** window, select the context table whose settings you want to view.

This opens the context table settings window. It displays the following information:

- **Name**—unique name of the resource.
- **Tenant**—the name of the tenant that owns the resource.
- **TTL**—the record added to the context table is stored in it for this duration. This value is specified in seconds.
- **Description**—any additional information about the resource.
- **Schema** is an ordered list of fields and their data types, with key fields marked.

Editing context table settings

To edit context table settings:

1. In the main menu, go to **Monitoring** → **Resources** → **Context tables**.

2. In the list in the **Context tables** window, select the context table whose settings you want to edit.

3. Specify the values of the following parameters:

   - **Name**—unique name of the resource.

   - **TTL**—the record added to the context table is stored in it for this duration. This value is specified in seconds.

   - **Description**—any additional information about the resource.

   - **Schema** is an ordered list of fields and their data types, with key fields marked. If the context table is not used in a correlation rule, you can edit the list of fields.

     If you want to edit the schema in a context table that is already being used in a correlation rule, follow the steps below.

   The **Tenant** field is not available for editing.

4. Click **Save**.

To edit the settings of the context table previously used by the correlator:

1. Export data from the table.

2. Copy and save the path to the file with the data of the table on the disk of the correlator. This path is specified in the **Directory** column in the **Correlator context tables** window. You will need this path later to delete the file from the disk of the correlator.

3. Delete the context table from the correlator.

4. Edit context table settings as necessary.

5. Delete the file with data of the table on the disk of the correlator at the path from step 2.

6. Add the context table in which you edited the settings to the correlator.

7. To restart the correlator, in the **Resources** → **Active services** section, in the list of services, select the check box next to the relevant correlator, click the three-dots icon on the toolbar and in the displayed menu, select **Restart**.

8. Adapt the fields in the exported table (see step 1) so that they match the fields of the table that you uploaded to the correlator at step 6.

9. Import the adapted data to the context table.

Duplicating context table settings

To copy a context table:

1. In the main menu, go to **Monitoring** → **Resources** → **Context tables**.
2. Select the check box next to the context table that you want to copy.
3. Click **Duplicate**.
4. Specify the necessary settings.
5. Click the **Save** button.

   The context table is copied.

Deleting a context table

You can delete only those context tables that are not used in any of the correlators.

To delete a context table:

1. In the main menu, go to **Monitoring** → **Resources** → **Context tables**.
2. Select the check boxes next to the context tables that you want to delete.
   To delete all context tables, select the check box next to the **Name** column.

   > At least one check box must be selected.

3. Click the **Delete** button.
4. Click **OK**.

   The context tables are deleted.

Viewing context table records

To view a list of context table records:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.

3. In the context menu of the correlator for which you want to view the context table, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. In the **Name** column, select the relevant context table.

The list of records for the selected context table is displayed.

The list contains the following data:

- **Key** is the composite key of the record. It is comprised by one or more values of key fields, separated by the "|" character. If one of the key field values is absent, the separator character is still displayed.

  For example, a record key consists of three fields: `DestinationAddress`, `DestinationPort`, and `SourceUserName`. If the last two fields do not contain values, the record key is displayed as follows: `43.65.76.98| | `.

- **Record repetitions** is the total number of times the record was mentioned in events and identical records were downloaded when importing context tables to KUMA.

- **Expiration date** – date and time when the record must be deleted.

  If the **TTL** field had the value of 0 when the context table was created, the records of this context table are retained for 36,000 days (approximately 100 years).

- **Updated** is the date and time when the context table was updated.

Searching context table records

To find a record in the context table:

1. In the main menu, go to **Monitoring** → **Resources**.

2. In the **Services** section, click the **Active services** button.

3. In the context menu of the correlator in whose context table you want to find a record, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. In the **Name** column, select your context table.

   This opens a window with the records of the selected context table.

5. In the **Search** field, enter the record key value or several characters from the key.

The list of context table records displays only the records whose key contains the entered characters.

If the your search query matches records with empty key values, the text <Nothing found> is displayed in the widget on the **Dashboard**. We recommend clarifying the conditions of your search query.

Adding a context table record

To add a record to the context table:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. In the context menu of the correlator to whose context table you want to add a record, select **Go to context tables**.

   This opens the **Correlator context tables** window.
4. In the **Name** column, select the relevant context table.

   The list of records for the selected context table is displayed.
5. Click **Add**.

   The **Create record** window opens.
6. In the **Value** field, specify the values for fields in the **Field** column.

   KUMA takes field names from the correlation rules with which the context table is associated. These names are not editable. The list of fields cannot be edited.

   If you do not specify some of the field values, the missing fields, including key fields, are populated with default values. The key of the record is determined from the full set of fields, and the record is added to the table. If an identical key already exists in the table, an error is displayed.

   List of default field values

Field type	Default value
Integer	0
Floating point number	0.0
String	""
Boolean	false
IP address	"0.0.0.0"
Timestamp	0
Integer list	[]
Float list	[]
List of strings	[]
Boolean list	[]
Timestamp list	[]
IP list	[]
7. Click the **Save** button.

The record is added.

Editing a context table record

To edit a record in the context table:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. In the context menu of the correlator for which you want to edit the context table, select **Go to context tables**.

   This opens the **Correlator context tables** window.
4. In the **Name** column, select the relevant context table.

   The list of records for the selected context table is displayed.
5. Click on the row of the record that you want to edit.
6. Specify your values in the **Value** column.
7. Click the **Save** button.

The record is overwritten.

Restrictions when editing a record:

- The value of the key field of the record is not available for editing. You can change it by exporting and importing a record.
- Field names in the **Field** column are not editable.
- The values in the **Value** column must meet the following requirements:
  - greater than or equal to 0 for fields of the **Timestamp** and **Timestamp list** types.
  - IPv4 or IPv6 format for fields of the **IP address** and **IP list** types.
  - is **true** or **false** for a **Boolean** field.

Deleting a context table record

To delete records from a context table:

1. In the main menu, go to **Monitoring** → **Resources**.
2. In the **Services** section, click the **Active services** button.
3. In the context menu of the correlator from whose context table you want to delete a record, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. In the **Name** column, select the relevant context table.

   The list of records for the selected context table is displayed.

5. Select the check boxes next to the records you want to delete.

   To delete all records, select the check box next to the **Key** column.

   At least one check box must be selected.

6. Click the **Delete** button.

7. Click **OK**.

The records will be deleted.

Importing data into a context table

To import data to a context table:

1. In the main menu, go to **Monitoring** → **Resources**.

2. In the **Services** section, click the **Active services** button.

3. In the context menu of the correlator to whose context table you want to import data, select **Go to context tables**.

   This opens the **Correlator context tables** window.

4. Select the check box next to your context table and click **Import**.

   This opens the context table data import window.

5. Click **Add** and select the file that you want to import.

6. In the **Format** drop-down list select the format of the file:

   - **csv**
   - **tsv**
   - **internal**

7. Click the **Import** button.

The data from the file is imported into the context table. Records that previously existed in the context table are preserved.

When importing, KUMA checks the uniqueness of each record's key. If a record already exists, its fields are populated with new values obtained by merging the previous values with the field values of the imported record.

If no record existed in the context table, a new record is created.

> Data imported from a file is not checked for invalid characters. If you use this data in widgets, widgets are displayed incorrectly if invalid characters are present in the data.

Exporting data from a context table

To export data from a context table:

1. In the main application window, go to the **Resources and services** → **Active services** section.

2. Select the correlator whose context table you want to export, and in the toolbar or in the menu (⋮), click **Go to context tables**.

   This opens the **Correlator context tables** window.

3. Select the check box next to the context table that you want to export.

   You cannot export multiple context tables at the same time. Each table must be exported separately.

4. Click the **Export** button and select a format from the **Format** drop-down list:

   - **internal**

     The context table is exported in JSON format.

   - **csv**

   - **tsv**

5. Click the **Export** button.

The context table is downloaded to your device in the selected format. The name of the exported file follows the following template: <context_table_name>_<export date (YYYY-MM-DD)>_<export time (HH-MM-SS)>. Exported data in the file is sorted by the **Key** field.

When exporting in JSON (**internal**) format:

- The order of the fields in the file is not defined.

- Fields of the array type are exported to a file as an array of values, for example: `["val1","val2","val3","val4"]`.

- The date and time format in the field of the **Timestamp** type depends on the selected application interface language:

  - Russian: DD.MM.YYYY HH:MM:SS, for example: `18.02.2025 16:09:58`

  - English: YYYY-MM-DD HH:MM:SS, for example: `2025.02.18 16:09:58`

When exporting in CSV and TSV formats:

- The first line of the file contains field names, and the subsequent lines contain the data.

- In the CSV format, the comma is used as a field separator, and in TSV format, the tabulation character.

You can also export data into the context table in the context table settings window.

Building a hierarchy of OSMP instances

You can arrange Open Single Management Platform (OSMP) instances into a two-level hierarchy, with the parent instance at the top. This enables you to distribute the settings from the parent instance to the child ones, while maintaining the settings of child instances.

For example, User A and User B deployed OSMP, and both the users have an independent instance of OSMP and work in their own instance of OSMP Console.

To enable data exchange between the instances, User A and User B have to arrange their instances into a hierarchy. For this, User A, the user of the future child instance, sends a connection request to the future parent instance; and user B, the user of the future parent instance, approves the request.

If necessary, you can disconnect the child instance.

After you build a hierarchy, the entities of the instances are managed in the hierarchy as follows:

- **Tenants**

  The tenant hierarchy of the child instance is built into the tenant hierarchy of the parent instance. Tenant settings and integration tenant settings of the child instance can be edited from both the child and the parent instance. Tenant settings and integration tenant settings of the parent instance can be edited from the parent instance only.

  On the parent instance, you can view the tenant hierarchy of the child instance; and create, edit, and delete tenants on the child instance; but you cannot delete the child instance Root tenant.

- **Assets (devices) and asset categories**

  Settings of the parent and the child instances are independent from each other. If you make changes in the parent instance, the changes are not applied to the child instance, and vice versa.

  On the child instance, the assets of the parent instance are not displayed, and the asset categories of the parent instance are available to view, to link to the child instance assets, and to run response actions for the child instance assets. For example, on both a parent and child instance there is an NNN asset category. If you run a response action for the NNN category from the child instance, only the child instance assets fall under the response action. However, you cannot edit the parent instance categories, assign these categories to the child instance devices, or create subcategories of the inherited categories.

  On the parent instance, asset categories of the child instance are displayed. If you select an asset category, the child instance assets are displayed. If you delete a category from the parent instance, then on the child instance this category becomes local and moved to the Categorized assets category on the child instance Root tenant. You cannot delete an asset category if it is used by KUMA resources. Assets categories of the child instance cannot be used by KUMA resources of the parent instance.

- **Users and user roles**

  All users are managed from the parent instance. Users and roles are replicated from the parent instance into the child one. In the child instance, existing users are deleted, and user roles remain without being replicated into the parent instance.

- **KUMA entities**

  - **Events**

    Child instance events are stored on the child instance, and parent instance events are stored on the parent instance. On the parent instance, you can perform a search of the child instances events.

  - **Correlation rules and IOA rules**

IOA rules are converted into KUMA correlation rules after the hierarchy is built.

There are the following modes of working with correlation rules in the hierarchy:

- **Local**. If the correlation rule is created on the child instance, it is managed on the child instance. If the correlation rule is inherited from the parent instance, then on the child instance you can only add exclusions to the correlation rule and cannot edit the rule.

  The correlation rules and their exclusions are only applied to the devices connected to the child instance Administration Server.

- **Global**. The correlation rules are created on the parent instance. The rules and their exclusions are applied to the child instance devices that you specified while configuring the rules settings.

- **Exclusions from correlation rules**

  Temporary exclusions added to correlation rules on the parent instance are not inherited. On the child instance, you can add a temporary exclusion to the inherited correlation rules. Filters are inherited with correlation rules.

- **Alerts and Incidents**

  Alerts and incidents of the child instance are replicated to the parent instance and displayed in both the child and the parent instance. Alerts and incidents of the parent instance are displayed in the parent instance only. On the parent instance, you cannot edit, link, and merge the child instance alerts or incidents.

  When deleting alerts or an incident on the child instance, they are deleted from the parent instance as well.

  Alerts and incidents settings, as well as their retention period, are not inherited. If the retention period expires, and an alert or incident is deleted on the child instance, the corresponding event is sent to the parent incident, and then the alert or incident is deleted on the parent instance as well.

- **Incident types**, **workflows**, **and SLA**

  To avoid data inconsistency, incident types, workflows, and SLA are copied to the child instance as a single bundle.

  - **SLA**

    On the child instance, you cannot create an SLA. When trying to view the child SLA from the parent instance, you are redirected to the child instance.

  - **Workflow**

    On the child instance, you can create new workflows, as well as view and apply the workflows inherited from the parent instance.When trying to view the child instance workflow from the parent instance, you are redirected to the child instance. On the parent instance, you can delete a parent instance workflow; if the workflow has no linked entities on the parent instance, the workflow is no longer displayed and after 90 days is completely deleted.You cannot delete a workflow with linked entities.

  - **Incident types**

    Incident types are inherited from the parent instance, where they have disabled status. If a default incident type is inherited, in the child instance this incident type is not default.

    On the child instance, you can create new incident types, as well as view and apply the inherited incident types. You cannot edit the inherited incident types. Also, you can use default the attribute both to the inherited incident types and ones created locally on the child instance.

- **Aggregation rules**

Manually created aggregation rules are inherited from the parent instance. Predefined aggregation rules are not inherited.

On the parent instance, you can view all aggregation rules of the child instance. The ability to edit the rules is defined by your XDR role.

On the child instance, the inherited aggregation rules are displayed at the top of the rules table in the order as they were displayed on the parent instance. By default, the rules are disabled when inherited. You can view, enable, and disable the inherited aggregation rules.

The enabled aggregation rules will be applied in accordance with their priority: the higher the rule is in the table, the higher its priority.

- **Segmentation rules**

  Manually created segmentation rules are inherited from the parent instance. Predefined segmentation rules are not inherited.

  On the parent instance, you can view all segmentation rules of the child instance. The ability to edit the rules is defined by your XDR role.

  On the child instance, the inherited segmentation rules are displayed at the top of the rules table in the order as they were displayed on the parent instance. By default, the rules are disabled when inherited. You can view, enable, and disable the inherited segmentation rules.

  The enabled segmentation rules will be applied in accordance with their priority: the higher the rule is in the table, the higher its priority.

- **YARA and IOC rules**

  The lists of YARA and IOC rules are inherited from the parent instance.

  On the parent instance, you can view and edit the child instance rules, as well as apply the child instance rules to the child instance assets. You cannot apply the child instance rules to the parent instance assets.

  On the child instance, you can view YARA and IOC rules inherited from the parent instance but you cannot edit these rules. To run response actions, you can use both the rules of the child instance and the ones inherited from the parent instance.

- **Sandbox rules**

  Sandbox rules are inherited from the parent instance.

  On the parent instance, you can view and edit the child instance Sandbox rules, as well as apply the child instance rules to the child instance assets.

  On the child instance, you can view the Sandbox rules inherited from the parent instance but you cannot edit these rules.

- **Licensing**

  License management and license settings are carried out by instances independently from each other and not inherited within the instances hierarchy.

  A license key is added to each instance separately. You can add the same license key to all the instances.

- **Policies and tasks**

  Policies and tasks are inherited from the parent instance.

- KSN/KPSN settings, credentials, and notifications templates are not inherited.

Connecting a child instance to the parent instance

The user of a future child instance sends a connection request, and the user of the future parent instance approves the request.

> Both users must have the Main Administrator XDR role in their instances.
> If you send a connection request to an instance that serves as a child instance in another hierarchy, the request will be canceled.
> The maximum number of child instances that a parent instance can have is 150.

Sending a connection request from the child instance

To send a connection request:

1. On the future child instance, in the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. In the toolbar, click the **Request connection** button.

3. In the right pane that opens, do the following:

   - In the **API URL of parent instance** field, specify the address of the parent instance public API, for example: `https://api.xdr-hier-1.kuma.example.com`.

   - In the **Thumbprint of the parent instance certificate** filed, specify the hash of the parent instance certificate.

     If the parent instance has a chain of certificates, we recommend that you specify an intermediate certificate. We do not recommend that you use the leaf certificate because it is rotated more often than an intermediate one. After the specified certificate is rotated, you need to restore the instance hierarchy.

   - In the **API token** field, specify the public API token for external integrations.

     > An API token must be previously generated on the parent instance.

   - If necessary, configure the following settings:

     - In the **API URL of child instance** field, specify the address of the child instance public API, for example: *https://api.xdr-hier-2.kuma.example.com.*

     - In the **Select tenants to share SIEM resources** section, select the check boxes next to the tenants whose SIEM resources you want to be visible in the parent instance.

4. Click the **Request** button.

   The pane closes.

5. In the confirmation dialog box, click **Request**.

   The confirmation dialog box closes, and the **Your connection request is pending approval** status message is displayed on the screen.

After the user on the parent instance approves or declines the request, you receive an email notification and the status message displayed on the screen also changes.

If the user on the parent instance approves the request, the ID of the parent instance and the name of the parent instance tenant are displayed in the child instance.

Approving a connection request from the parent instance

Once the user of the child instance sends a connection request, you receive a corresponding email notification.

To approve the connection request:

1. In the main menu, go to **Settings** → **Tenants**.

2. At the top of the page, click the **View requests** link that is displayed in the message about pending connection requests.

3. In the **Connection requests** pane that opens, click the **Approve** button for a request that you want to approve.

4. In the **Approve connection request** pane that opens, do the following:

   - In the **Root tenant name** field, specify a new name for the child instance tenant to be connected.

   - In the **Root tenant name** section, from the tenant hierarchy of the parent instance, select a tenant to be assigned as a parent for the tenant of child instance.

   - Click the **Approve** button.

     The pane closes.

   The child instance is added to the tenant hierarchy of the parent instance. A corresponding message is displayed on the screen.

Restarting the active services on the child instance

After the connection between the child and parent instances is established, restart the active services on the child instance. After that, the active services of the child instance are displayed on the parent instance.

Results

You can now manage the tenant tree of the child instance as follows:

- View the child instance tenant tree
- Create tenants on the child instance
- Edit tenants on the child instance
- Delete tenants from the child instance except for the Root tenant
- Disconnect the child instance

The user of the child instance can view, edit, and delete tenants of the child instance only.

Disconnecting a child instance from a parent instance

After a child instance is disconnected from its parent instance, you cannot connect the child instance to another hierarchy.

Initiating disconnection from a parent instance

To disconnect a child instance:

1. On the parent instance, in the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Select the check box next to the child instance Root tenant to be disconnected.

3. In the toolbar, click the **Disconnect** button, and then in the confirmation dialog box that is displayed, click **Disconnect**.

   The child instance is disconnected from the parent instance, and an appropriate message is displayed on the screen. The user of the child instance will be notified by email.

If a child instance is turned off at the moment of disconnection, the instance will be disconnected forcibly. Anyway, to complete the disconnection, turn on the child instance, and then follow the instructions below to disconnect the child instance from the parent one.

When you disconnect instances, Kaspersky Next XDR Expert creates an audit event `Disconnecting an SCN from PCN`. For both cases—when the child instance is turned on or turned off at the moment of disconnection, the audit event is the same.

For diagnostics purposes, if a child instance is turned off at the moment of disconnection, Kaspersky Next XDR Expert adds an error to the hierman component log files, for example the "disconnect on secondary failed" error. Use the value of the `trace_id` field in the audit event to find the errors in the log files.

Initiating disconnection from a child instance

To disconnect a child instance:

1. On the child instance, in the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Select the check box next to the Root tenant to be disconnected.

3. In the toolbar, click the **Disconnect** button, and then in the confirmation dialog box that is displayed, click **Disconnect**.

The child instance is disconnected from the parent instance, and an appropriate message is displayed on the screen. The user of the parent instance will be notified by email.

Disconnection results

As a result of the child instance disconnection, the OSMP hierarchy is broken.

In the parent instance:

- The resources and settings of the disconnected instance are deleted.

- The child instance data replicated to the parent instance is now deleted from the parent instance.

In the child instance:

- Information about the parent instance is deleted, and the parent instance is not displayed in the child instance tenant hierarchy.

- Management of the child instance tenant hierarchy goes back to the child instance.

- The Identity and Access Manager remains available in the read-only mode.

  The users of the child instance are still able to sign in to the instance, but cannot perform any other actions (create users, assign rights, change a password, etc.).

  The users created on the child instance before the instance was connected to the hierarchy are deleted from the child instance.

- The entities inherited from the parent instance are immediately deleted from the child instance if the entities have no linked objects on the parent instance. Otherwise, the entity is no longer displayed on the child instance but remains available in the read-only mode for 90 days. After this period, the entity is deleted.

- Asset categories inherited from the parent instance are now managed on the child instance.

- The services managed by the parent instance are deleted.

Integration with other solutions

Integration with other solutions allows you to enrich the functionality of Kaspersky Next XDR Expert.

Kaspersky Next XDR Expert supports integration with the following Kaspersky and third-party solutions:

- Kaspersky Automated Security Awareness Platform
- Kaspersky Threat Intelligence Portal
- Kaspersky Anti-Targeted Attack Platform / Kaspersky Endpoint Detection and Response
- Active Directory
- UserGate
- Ideco NGFW
- Ideco UTM
- Redmine
- Check Point NGFW
- Sophos Firewall
- Continent 4
- SKDPU NT
- FortiGate
- Palo Alto NGFW

Kaspersky Next XDR Expert also supports more than 100 event sources. For the full list of supported event sources, refer to the Supported event sources section.

Integration settings can be specified for a tenant of any level. Parent integration settings are copied to a child tenant. You can edit the copied child integration settings, since child and parent settings are not related and changes in child settings do not affect the settings in the parent tenant.

> For the shared tenant, you do not need to configure the integration settings.

If you need to disable integration, you can do it manually in the **Settings** → **Tenants**.

> Integration with a Kaspersky solution is removed automatically when the tenant for which the integration was specified is removed. The delay when removing data is up to 24 hours. Restoring integration settings is not available.

Integration with Kaspersky Automated Security Awareness Platform

Kaspersky Automated Security Awareness Platform (hereinafter also referred to as KASAP) is an online learning platform that allows users to learn the rules of information security and related threats in their daily work, as well as to practice with real examples.

After configuring integration, you can perform the following tasks in Kaspersky Next XDR Expert:

- Assign learning courses to users who are associated with alerts and incidents.
- Change user learning groups.
- View information about the courses taken by the users and the certificates they received.

KASAP is considered to be integrated with Kaspersky Next XDR Expert after the integration between KASAP and KUMA is configured.

Before configuring integration between KASAP and KUMA, you need to create an authorization token and obtain a URL for API requests in KASAP.

Creating a token in KASAP and getting a URL for API requests

Creating a token

To authorize API requests from KUMA to KASAP, the requests must be signed with a token created in KASAP.

Only the company's administrator can create a token.

To create a token:

1. Sign in to the KASAP web interface.
2. In the **Dashboard** section, select the **Import and sync** section, and then open the **OpenAPI** tab.
3. Click the **New token** button.
4. In the window that opens, select the token rights available during integration:
   - GET /openapi/v1/groups
   - POST /openapi/v1/report
   - PATCH /openapi/v1/user/:userid

5. Click the **Generate token** button.

   The generated token is displayed on the screen.

6. Copy the token and save it in any convenient way. This token is required to configure integration between KASAP and KUMA.

> The token is not stored in the KASAP system in the open form. After you close the **Create token** window, the token is unavailable for viewing. If you close the window without copying the token, you will need to click the **Reissue token** button for the system to generate a new token.

The issued token is valid for 12 months.

Getting a URL for API requests

The URL is used for interacting with KASAP via OpenAPI. You have to specify this URL when configuring integration between KASAP and KUMA.

To get the URL used in KASAP for API requests:

1. Sign in to the KASAP web interface.

2. In the **Dashboard** section, select the **Import and sync** section, and then open the **OpenAPI** tab.

3. In the **OpenAPI URL** field, copy the URL, and then save it in any convenient way.

Integration with Kaspersky Threat Intelligence Portal

You must configure integration with Kaspersky Threat Intelligence Portal (hereinafter also referred to as Kaspersky TIP) to obtain information about the reputation of the observable objects.

> Before configuring the settings, you have to create an authorization token for API requests on Kaspersky TIP or Kaspersky OpenTIP.

To configure integration between Kaspersky Next XDR Expert and Kaspersky TIP:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then select the **Kaspersky TIP** section.

> You can edit the **Kaspersky TIP** section if you are assigned one of the following XDR roles: Main administrator, Tenant administrator, or SOC administrator.

4. If at step 2 you selected the Root tenant, you can turn on the **Proxy** toggle button to use a proxy server for interaction with Kaspersky TIP.

   The proxy server is configured in the root Administration Server properties.

5. In the **Cache TTL** field, specify the period of cache storage and the units: days or hours.

   By default, 7 days is set. If you do not specify any value, the period of cache storage is unlimited.

   > You set the period of cache storage for all connections.

6. Turn on the **Integration** toggle button for one of the following services:

   - Kaspersky TIP (General access)

     After you add an authorization token, you will be able to obtain information from Kaspersky TIP about the following types of observables listed at the **Observables** tab in the alert or incident details: domain, URL, IP, MD5, SHA256. The information is updated in the **Enrichment** column. Quota is consumed when you request data.

   - Kaspersky TIP (Premium access)

     After you add an authorization token, you will be able to do the following:

     - Obtain information from Kaspersky TIP about the following types of observables listed at the **Observables** tab in the alert or incident details: domain, URL, IP, MD5, SHA256. The information is updated in the **Enrichment** column. Quota is consumed when you request data.

     - Obtain information from Kaspersky TIP about the following types of observables listed at the **Observables** tab in the alert or incident details: domain, URL, IP, MD5, SHA256. The information is updated in the **Status update** column. Quota is not consumed when you request data.

7. Click the **Add token** button.

8. In the window that opens, enter the authorization token, and then click the **Add** button.

   > For details about generating an authorization token for API requests, refer to the Kaspersky TIP or Kaspersky OpenTIP help.

   After you add the token, you can change it by clicking the **Replace** button, and then entering a new token in the window that opens. This may be necessary if the token is expired.

9. Click the **Save** button.

Integration with KATA/KEDR

Kaspersky Endpoint Detection and Response (hereinafter also referred to as KEDR) is a functional block of Kaspersky Anti Targeted Attack Platform (hereinafter also referred to as KATA) that protects assets in an enterprise LAN.

You can configure integration between Kaspersky Next XDR Expert and KATA/KEDR to manage threat response actions on assets connected to Kaspersky Endpoint Detection and Response servers. Commands to perform operations are received by the Kaspersky Endpoint Detection and Response server, which then relays those commands to Kaspersky Endpoint Agent installed on assets.

To configure integration between Kaspersky Next XDR Expert and KATA/KEDR:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then select the **KATA/KEDR** section.

   > You can edit the **KATA/KEDR** section, if you are assigned one of the following XDR roles: Main administrator, Tenant administrator or SOC administrator.

4. Turn on the **KATA integration** toggle button.

5. Click the **Add connection** button, and then in the window that opens do the following:

   a. In the **IP address or host name** field, enter one of the following:

      - hostname
      - IPv4
      - IPv6

   b. In the **Port** field, set a port.

   c. Click the **Save** button.

   The window is closed.

   If the connection is not added, an error message is displayed.

   If the connection is added successfully, an appropriate message is displayed on the screen. An XDR ID, certificate, and private key are generated and displayed in the corresponding fields. If necessary, you can generate the new certificate and private key by clicking the **Generate** button.

   To ensure that the connection is established successfully, click the **Check connection** button. The result is displayed in the **Connection status** parameter.

6. Click the **Save** button to save the settings.

   After you add the connection, you can edit or delete it by clicking the corresponding icons. You can also add another connection by performing steps 1–6.

If you want to receive information about Kaspersky Endpoint Detection and Response alerts, you need to configure integration between the KUMA component and KATA/KEDR.

Configuring integration between Kaspersky Next XDR Expert and script launch service

You can respond to alerts and incidents via external systems by launching third-party scripts on remote client devices. To enable this option, you have to configure the environment and integration between Kaspersky Next XDR Expert and the script launch service.

To configure environment for launching third-party custom scripts, you must:

- Set a device on which the third-party custom script is launched.
- Configure integration between Kaspersky Next XDR Expert and the script launch service.
- Create a playbook that will be used to launch the script.

It is the customer who provides access to third-party custom scripts and updates the scripts.

To configure integration between Kaspersky Next XDR Expert and the script launch service:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then in the **Custom integration** section:

   - Turn on the **Custom integration** toggle button.
   - In the **Remote host verification** section, turn on the **Verify the host before connecting** toggle button, and then fill in the **Public key** field to enable verification of a client device in Kaspersky Next XDR Expert.
   - In the **Remote host connection** section, do the following:
     - Fill in the **IP address or host name** and **Ports** fields.
     - Select an **SSH authentication method** that will be used to establish a secure connection with a remote device:
       - **User name and password**. If you select this authentication method, at the next step you must enter the user name and password.
       - **SSH key**. If you select this authentication method, at the next step you must enter the user name and SSH key.
     - Click the **Add data** button.

4. In the window that opens, enter the required data, and then click the **Save** button.

If you want to edit the data you saved, click the **Replace** button, enter new data in the window that opens, and then save the edits.

To ensure that the connection is established successfully, click the **Check connection** button. The result is displayed in the **Connection status** parameter.

5. Click the **Save** button to save the settings.

Integration between Kaspersky Next XDR Expert and the script launch service is configured. You can perform response actions on remote devices by launching playbooks.

Threat detection

Open Single Management Platform uses alerts and incidents as work items that are to be processed by analysts.

The **Alerts** and **Incidents** sections are displayed in the main menu if the following conditions are met:

- You have a license key for Kaspersky Next XDR Expert.
- You are connected to the root Administration Server in OSMP Console.
- You have one of the following XDR roles: Main administrator, Tenant administrator, Junior analyst, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Auditor.

Working with alerts

This section contains general information about alerts, their properties, typical life cycle, and connection with incidents. The instructions that are provided will help you analyze the alert table, change alert properties according to the current state in the life cycle, and combine alerts into incidents by linking or unlinking the alerts.

The **Alerts** section is displayed in the main menu if the following conditions are met:

- You have a license key for Kaspersky Next XDR Expert.
- You are connected to the root Administration Server in OSMP Console.
- You have one of the following XDR roles: Main administrator, Tenant administrator, Junior analyst, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Auditor.

About alerts

An *alert* is an event in the organization's IT infrastructure that was marked by Open Single Management Platform as unusual or suspicious, and that may pose a threat to the security of the organization's IT infrastructure.

Open Single Management Platform generates an alert when an EPP application (for example, Kaspersky Endpoint Security for Windows) detects certain activity in the infrastructure that corresponds to conditions defined in the detection rules.

The alert is created within 30 seconds after the KUMA correlation event has occurred. You receive a corresponding email notification if you previously configured notifications templates.

You can also create an alert manually from a set of events.

After detection, Open Single Management Platform adds alerts to the alert table as work items that are to be processed by analysts. You cannot delete alerts—you can only close them.

Alerts can be assigned only to analysts who have the access right to read and modify alerts and incidents.

You can manage alerts as work items by using the following alert properties:

- **Alert status**

  Possible values: **New**, **In progress**, **Closed**, or **In incident**.

  The alert status shows the current state of the alert in its life cycle. You can change the status as you like, with the following exceptions:

  - You cannot return closed alerts to the status **In progress**. Closed alerts can only be returned to the status *New*, and then the status can be changed to **In progress**.
  - You cannot set the **In incident** status manually. The alerts gain this status when they are linked to an incident.
  - You can only set the **Closed** status to a linked alert. To set the **New** or **In progress** status, you first must unlink the alert from the incident.

- **Alert severity**

  Possible values: **Low**, **Medium**, **High**, or **Critical**.

  The alert severity shows the impact this alert may have on computer security or corporate LAN security, based on Kaspersky experience. The severity is defined automatically and cannot be changed manually.

- **Alert assignee**

  This is an alert owner, the analyst who is responsible for the alert investigation and process. You can change an alert assignee at any time, with one exception—you cannot change an assignee of closed alerts.

You can combine and link alerts to bigger work items called *incidents*. You can link alerts to incidents manually, or enable the rules to create incidents and link alerts automatically. By using incidents, analysts can investigate multiple alerts as a single issue. When you link a currently unlinked alert to an incident, the alert loses its current status and gains the status **In incident**. You can link a currently linked alert to another incident. In this case, the **In incident** status of the alert is kept. You can link a maximum of 200 alerts to an incident.

Each alert has *alert details* that provide all of the information related to the alert. You can use this information to investigate the alert, track the events that preceded the alert, view detection artifacts, affected assets, or link the alert to an incident.

Alert data model

The structure of an alert is represented by fields that contain values (see the table below). Some fields are objects or arrays of objects with their own set of fields (for example, the `Assignee` and `Assets` fields).

Alert

Field	Value type	Is required	Description
`InternalID`	String	Yes	Internal alert ID, in the UUID format. The field value may match the `SourceID` value.

`ID`	Integer	Yes	Short internal alert ID.
`TenantID`	String	Yes	ID of the tenant that the alert is associated with, in the UUID format.
`CreatedAt`	String	Yes	Date and time of the alert generation, in the RFC 3339 format.
`UpdatedAt`	String	Yes	Date and time of the last alert change, in the RFC 3339 format.
`StatusChangedAt`	String	No	Date and time of the last alert status change, in the RFC 3339 format.
`Severity`	String	Yes	Severity of the alert. Possible values: • `critical` • `high` • `medium` • `low`
`IntegrationID`	String	Yes	ID of the Kaspersky application management plug-in that is integrated in OSMP.
`IntegrationCompatibilityVersion`	String	Yes	Version of the Kaspersky application management plug-in that is integrated in OSMP.
`SourceID`	String	No	Unique alert identifier in the integrated component.
`SourceCreatedAt`	String	No	Date and time of the alert generation in the integrated component, in the RFC 3339 format.
`FirstEventTime`	String	Yes	Date and time of the first telemetry event related to the alert, in the RFC 3339 format.
`LastEventTime`	String	Yes	Date and time of the last telemetry event related to the alert, in the RFC 3339 format.
`DetectSource`	String	No	Component that detects and generates the alert.
`DetectionTechnologies`	Array of strings	No	Triggered detection technology.
`Status`	String	Yes	Alert status. Possible values: • `new` • `inProgress` • `inIncident` • `closed`
`StatusResolution`	String	No	Resolution of the alert status. Possible values: • `truePositive` • `falsePositive` • `lowPriority` • `merged`
`IncidentID`	String	No	Internal ID of the incident associated with the alert.
`IncidentLinkType`	String	No	Way to add an alert to an incident. Possible values: • `manual` • `auto`
`Assignee`	`Assignee` object	No	Operator to whom the alert is assigned.
`MITRETactics`	Array of `MITRETactic` objects	No	MITRE tactics related to all triggered IOA rules in the alert.
`MITRETechniques`	Array of `MITRETechnique` objects	No	MITRE techniques related to all triggered IOA rules in the alert.
`Observables`	Array of `Observable` objects	No	Observables related to the alert.
`Assets`	Array of `Asset` objects	No	Assets affected by the alert.

`Rules`	Array of `Rule` objects	No	Triggered correlation rules, on the basis of which the alert is generated.
`OriginalEvents`	Array of objects	No	Events, on the basis of which the alert is generated.
`ExternalRef`	String	Yes	Link to an entity in an external system (for example, a link to a Jira ticket).
`Extra`	Object	No	Data related to the alert, in the JSON format. This data is obtained from managed Kaspersky applications when events are transformed into alerts. This field is not used in the interface.
`AdditionalData`	Object	No	Additional information about the alert, in the JSON format. This information can be filled in by a user or a playbook.
`Name`	String	Yes	Alert name.
`Attachments`	Array of `UnkeyedAttachment` objects	No	Attachments related to the incident.

Assignee

Field	Value type	Is required	Description
`ID`	String	Yes	User account ID of the operator to whom the alert is assigned.
`Name`	String	Yes	Name of the operator to whom the alert is assigned.

MITRETactic

Field	Value type	Is required	Description
`ID`	String	Yes	ID of the MITRE tactic related to all triggered IOA rules in the alert.
`Name`	String	Yes	Name of the MITRE tactic related to all triggered IOA rules in the alert.

MITRETechnique

Field	Value type	Is required	Description
`ID`	String	Yes	ID of the MITRE technique related to all triggered IOA rules in the alert.
`Name`	String	Yes	Name of the MITRE technique related to all triggered IOA rules in the alert.

Observable

Field	Value type	Is required	Description
`Type`	String	Yes	Type of the observable object. Possible values: • `ip` • `md5` • `sha256` • `url` • `domain` • `userName` • `hostName`

`Value`	String	Yes	Value of the observable object.
`Details`	String	No	Additional information about the observable object.

Rule

Field	Value type	Is required	Description
`ID`	String	Yes	ID of the triggered rule.
`Name`	String	No	Name of the triggered rule.
`Severity`	String	No	Severity of the triggered rule. Possible values: • `critical` • `high` • `medium` • `low`
`Confidence`	String	No	Confidence level of the triggered rule. Possible values: • `high` • `medium` • `low`
`Custom`	Boolean	No	Indicator that the alert is based on custom rules.

Asset

Field	Value type	Is required	Description
`Type`	String	Yes	Type of the affected asset (a device or an account). Possible values: • `host` • `user`
`ID`	String	Yes	ID of the affected asset (a device or an account).
`Name`	String	No	The name of the affected device that the alert is associated with (if `Type` is set to `host`). The user name of the affected user account associated with events, on the basis of which the alert is generated (if `Type` is set to `user`).
`IsAttacker`	Boolean	No	Indicator that the affected asset (a device or an account) is an attacker.
`IsVictim`	Boolean	No	Indicator that the affected asset (a device or an account) is a victim.

UnkeyedAttachment

Field	Value type	Is required	Description
`AttachmentID`	String	Yes	Attachment ID, in the UUID format.
`Name`	String	Yes	Attachment name.
`CreatedAt`	String	Yes	Date and time of the attachment creation, in the UTC format.
`UpdatedAt`	String	Yes	Date and time of the last attachment change, in the UTC format.
`CreatedBy`	String	Yes	Indicator that the affected asset (a device or an account) is a victim.
`Size`	Integer	Yes	Attachment size, specified in bytes.

`Status`	String	Yes	Attachment status that indicates whether the attachment upload is in progress, completed, or aborted with an error. Possible values: • `completed` • `error` • `uploading`
`Description`	String	No	Attachment description.
`StatusCode`	String	No	Text of the status that is displayed to a user (for example, an error message that is displayed when the attachment upload fails).

Viewing the alert table

The alert table is displayed in the **Monitoring** → **Alerts** section and provides you with an overview of all alerts registered by Open Single Management Platform. By default, the table displays the alerts related to all of the tenants to which you have access rights. If necessary, you can configure the alert table.

To configure the alert table, do any of the following:

- Apply tenant filter:

  a. Click the link next to the **Tenant filter** setting.

  The tenant filter opens.

  b. Select the check boxes next to the required tenants.

- Filter the data of the alerts table using one of the following methods:

  - Use the main filter:

    a. Click the filter ( ) icon.

    b. On the **Filters** tab, specify and apply the filter criterion in the invoked menu.

  - Filter out rescanned alerts:

    a. Click the header of the **Name** column.

    b. Check either **Only rescanned alerts** or **Exclude rescanned alerts** box.

  - Use a cell context filter to quickly add a new filter condition:

    a. Find the cell that contains the value that you want to add as a filter condition. Note that some columns do not have a cell context filter.

    b. Left-click the value in the cell.

c. In the list that opens, select one of the following actions:

- **Add to filter** to add the value in the filter condition.
- **Exclude from filter** to exclude the value from the filter condition.
- **Copy** to copy the value to the clipbоard.

- If you want to hide or display a column, click the settings (⚙) icon, and then select the necessary column.

The alert table is configured and displays the data you need.

The alert table has the following columns:

- **Alert ID**. The unique identifier of an alert.
- **Registered**. The date and time when the alert was added to the alert table.
- **Updated**. The date and time of the last change from the alert history.
- **Status**. The current status of the alert.
- **Assigned to**. The current assignee of the alert.
- **Tenant**. The name of the tenant in which the alert was detected.
- **Technology**. Names of the modules or components that generated the alert while scanning.
  Possible values in this column:
  - AM (Anti-Malware Engine)
  - YARA
  - SB (Sandbox)
  - IOA
  - IOC
- **Details**. Brief summary of the alert.
- **Detected**. One or multiple categories of detected objects. For example, when Kaspersky Next XDR Expert detects a file infected with the Trojan-Downloader.JS.Cryptoload.ad virus, the **Detected** field shows the Trojan-Downloader.JS.Cryptoload.ad category for this alert.
- **Rules**. The IOC or IOA rules that were triggered to detect the alert.
- **Affected assets**. The devices and users that were affected by the alert.
- **Number of assets**. Total number of affected assets.
- **Observables**. Detection artifacts, for example IP addresses or MD5 hashes of files.
- **Number of observables**. Total number of detection artifacts.

- **Incident link type**. Way to add an alert to an incident.
- **Severity**. Severity of the alert.
- **Name**. The name of the alert.
- **Status changed**. The date and time of the last alert status change.

Special considerations for event aggregation in alerts triggered by IOA rules

If Kaspersky Next XDR Expert has created an alert triggered by an IOA rule, and if the same IOA rule is triggered by other events on this or other hosts during the same day, Kaspersky Next XDR Expert aggregates those events in the same alert. If the alert is already closed, Kaspersky Next XDR Expert reopens it. The event aggregation period ends at 00:00 the next day.

Viewing alert details

Alert details are a page in the interface that contains all of the information related to the alert, including the alert properties.

To view alert details:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. In the alert table, click the ID of the required alert.

   The alert details are displayed.

If necessary, you can refresh the information in the alert details by clicking the refresh (⟲) icon next to the alert name.

The toolbar in the upper part of the alert details allows you to perform the following actions:

- Edit the **External reference** field value
- Assign the alert to an analyst
- Change the alert status
- Link the alert to an incident
- Unlink the alert from the incident
- Select a playbook
- Create a new incident and link the alert to it

Depending on alert type, alert details contain the following sections:

- **Summary**

  Depending on alert type, the summary section may contain the following alert properties:

  - **Analyst**. An analyst to which the alert is assigned.
  - **Name**. The name of the alert. If the alert name is not specified, this field is not displayed.
  - **Tenant**. The name of the tenant in which the alert was detected.
  - **Assets**. The number of user accounts and devices related to the alert.
  - **Severity**. Possible values: **Low**, **Medium**, **High**, or **Critical**. The alert severity shows the impact this alert may have on computer security or corporate LAN security based on Kaspersky experience.
  - **Rules**. The rules that were triggered to detect the alert. By clicking the ellipsis icon next to the rule name, you can open the context menu. Use this menu to learn more details about the rule, find alerts or incidents that were detected by the same rule, or search the rule triggering events in **Threat hunting** for the period between the first and the last event of the alert.

    When you click **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

  - **Registered**. A date and time when the alert was added to the alert table.
  - **First event**. A date and time of the first event related to the alert.
  - **Last event**. A date and time of the most recent event related to the alert.
  - **External reference**. Link to an entity in an external system (for example, a link to a Jira ticket). You can click the **Edit** button at the top to specify the external reference. If the alert was created from a Kaspersky Endpoint Detection and Response alert, a link to this alert is displayed in the field.
  - **Linked to**. The incident to which the alert is linked.
  - **MITRE tactic**. A tactic or several tactics detected in the alert. The tactics are defined in the MITRE ATT&CK knowledge base.
  - **MITRE technique**. A technique or several techniques detected in the alert. The techniques are defined in the MITRE ATT&CK knowledge base.
  - **Correlation rules**. The correlation rules that were triggered to detect a threat and create the alert.
  - **IOA rules**. The IOA rules that were triggered to detect a threat and create the alert. You can open the context menu by clicking an IOA rule name. By using the context menu, you can do the following:
    - View the detailed information about that IOA rule by clicking the **Open properties** button.
    - View the related events by clicking the **Go to Threat hunting** button.
    - Find similar incidents.

- View the detailed information about an IOA rule in Kaspersky TIP by clicking the **Find in Kaspersky TIP** button.
- Copy the IOA rule name.

- **Technology**. Detection technology used to detect a threat.
- **Detection source**. The source from which the detected object came. For example, if a file is sent to Sandbox for scanning from some host, this host is the detection source.
- **Additional data**. Additional information on the alert. You can edit a value in this field only by using a playbook. The field is displayed if you added a value.

- **Scan result**

The **Scan result** section contains information about the file that was scanned and the scan results that triggered the alert creation.

- **IOC scan results**

The **IOC scan result** section contains information about IOC rules that were triggered during the scan.

By clicking the IOC rule name, you can view the detailed information about the IOC rule.

By clicking **View report** in the **Scan results** column, you can view the detailed information about scan results, including network and system information, processes and files involved when scanning.

- **Kaspersky Sandbox scan results**

The **Kaspersky Sandbox scan result** section contains information about the results of processing objects in Sandbox including the following:

- Number of detected objects and their types.
- Number of extracted files and their types.
- Hash sums of the detected objects.

By clicking **Export results**, you can view the detailed information about the results of object processing in operating systems available on Sandbox, including the following:

- Diagrams with detection statistics.
- A graph containing a tree of object activity.
- MITRE tactics and techniques (see the MITRE ATT&CK knowledge base).
- A list of suspicious activities in an operating system.
- Screenshots of the file execution in an operating system.
- System activities.
- Network activities.

To get the full information in the Sandbox scan results window, switch between operating systems and between the results obtained in **Quick scan mode** and **Full logging mode**.

- **YARA scan results**

  The **YARA scan result** section contains information about the results of YARA scanning on a host.

  By clicking **View report** in the **Scan results** column, you can view the detailed information about scan results. To get the full information in the YARA scan results window, switch between the **Processes & memory dump**, **Files** and **Rule condition** tabs.

- **Scanned files**

  In the **Scanned files** section, you can find information about the files that were scanned. To learn more about threats detected in files on Kaspersky Threat Intelligence Portal, click the threat name in the **Verdict** column.

- **Details**

  In the **Details** section, you can track the telemetry events related to the alert.

  The event table displays the search result that you define through an SQL query.

  The toolbar of the event table allows you to perform the following actions:

  - **Download events**. You can click this button to download information about related events as a CSV file (in UTF-8 encoding).

  - **Find in Threat hunting**. You can click this button to open the **Threat hunting** section. This section allows you to search through all of the events related to the tenants that you have access to, and not only the events related to the current alert. By default, the opened event table contains all of the events that occurred during the time period between the first and the last event of the alert. For example, you can run a search query to find all of the events in which the device was affected.

    In the **Threat hunting** section, you can link events to alerts manually. This might be helpful if you discover that some events relate to an alert, but they were not linked to the alert automatically. For details, refer to the instructions on linking or unlinking events to or from alerts.

    You can go back to the incident details by clicking **Alert investigation** or by clicking the back button in your browser.

  - **Unlink from alert**. You can select an event or several events in the table, and then click this button to unlink the selected events from the alert.

- **Assets**

  In the **Assets** section, you can do the following:

  - View the devices and users affected by or involved in the alert.

  - Perform response actions. To do this, select the check box next to the asset, and then in the **Select response action** drop-down list, select the required response action.

  - Export information about affected assets to a CSV file by clicking the **Export to CSV** button.

  - Connect to the asset by using the remote terminal. To do this, click the **Open remote terminal** button.

The asset table contains the following columns:

- **Asset type**

  Possible values: device or user.

- **Asset name**

- **Asset ID**

- **Has signs of**

  Possible values: attacker or victim.

- **Authorization status**

  This parameter is only applied to device asset type. A device authorization status is defined by KICS for Networks. You can change the authorization status by applying the corresponding response action to a device.

- **Administration Server**

  The Administration Server that manages the device.

- **Administration Group**

  The administration group to which the device belongs.

- **Categories**

  Asset categories which include the asset.

By clicking a user name or a device name, you can:

- View the user or the device properties.

  > To view the list of alerts or incidents related to the device, in the device properties window, click the **Show related** button and select either **Alerts** or **Incidents**, depending on what you want to view.

- Copy the user name or the device name in the clipboard.

- Search the user name or the device ID in other alerts.

- Search the user name or the device ID in other incidents.

- Search the user name or the device ID in **Threat hunting** for the period between the first and the last event of the alert.

  > When after clicking a user name or a device name you select **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

By clicking a user ID or a device ID, you can:

- View the user or the device properties.

  To view the list of alerts or incidents related to the device, in the device properties window, click the **Show related** button and select either **Alerts** or **Incidents**, depending on what you want to view.

- Copy the user ID or the device ID in the clipboard.
- Search the user ID or the device ID in other alerts.
- Search the user ID or the device ID in other incidents.
- Search the user ID or the device ID in **Threat hunting** for the period between the first and the last event of the alert.

  When after clicking a user ID or a device ID you select **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

- **Files**

The **Files** section contains files obtained through response actions or manually added to this alert. You can upload, download, delete files related to the alert, or edit the files' description.

You can upload files of any extension. Duplicate file names are allowed. The maximum number of files that you can attach to the incident and the maximum total file size you can specify in the configuration file.

Default limitations:

- The number of files cannot exceed 100 per alert.
- Total file size cannot exceed 25 MB per alert.

To upload files, click the **Upload** button and select one or multiple files. If you attempt to upload files exceeding the limitations, the **Uploading files** panel displaying a warning message will open. In this panel, you can remove files from the upload queue until the warning message disappears and click the **Upload** button to upload files. If you click the **Upload** button ignoring the warning message, upload will fail and the file list will include files that could not be uploaded with a warning icon next to the file names.

Click a file to open the **Edit file** panel that displays file details. In this panel, you can edit file description.

Use check boxes to select a file or multiple files. Select one or more files and click the **Download** button to download them as an archive in the ZIP format. The archive is protected with the password *infected*. Select a file or multiple files and click the **Delete** button to delete the selected files.

The **Write** permission in the **Alerts and incidents** functional area is required to upload and delete files and edit file descriptions. The **Read** permission in the **Alerts and incidents** functional area is required to download files.

- **Observables**

In the **Observables** section, you can view the observables related to the alert.

You also can export information about the related observables to a CSV file by clicking the **Export to CSV** button.

The observables may include:

- MD5 hash
- URL
- Domain name
- SHA256
- UserName
- HostName
- Local IP address
- Remote IP address

  The following private IP addresses are not included in the observables related to the alert:
  From 10.0.0.0 to 10.255.255.255 with the subnet mask 255.0.0.0 or /8
  From 172.16.0.0 to 172.31.255.255 with the subnet mask 255.240.0.0 or /12
  From 192.168.0.0 to 192.168.255.255 with the subnet mask 255.255.0.0 or /16
  From 100.64.0.0 to 100.127.255.255 with the subnet mask 255.192.0.0 or /10

  To include private IP addresses in the observables, run the following commands on the administrator host:

  ```
  k0s kubectl -n irp set env deploy/transformer -c transformer
  OSMP_TRANSFORMER_FILTER_PRIVATE_IPS_DISABLED=true
  ```

  ```
  k0s kubectl -n irp set env deploy/incident -c incident
  OSMP_FILTER_PRIVATE_IPS_DISABLED=true
  ```

- Local Port
- Remote Port
- System Process ID
- Unique PID
- Full name of a file

By clicking a link in the **Value** column, you can:

- Search the observable value in **Threat hunting** for the period between the first and the last event of the alert.

  When after clicking a link in the **Value** column you select **Go to Threat hunting** in the menu, the **Threat hunting** section opens on the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

- Search the observable value in other alerts.
- Search the observable value in other incidents.
- Copy the observable value in the clipboard.

The toolbar of this section contains the following buttons:

- **Request status from Kaspersky TIP**. Use this button to obtain detailed information about the selected observable from Kaspersky Threat Intelligence Portal (Kaspersky TIP). As a result, the information is updated in the **Status update** column. Requires integration with Kaspersky Threat Intelligence Portal (Premium access).
- **Enrich data from Kaspersky TIP**. Use this button to obtain detailed information about all of the listed observables from Kaspersky TIP. As a result, the information is updated in the **Enrichment** column. Use a link in the **Enrichment** column to open the obtained enrichment details about an observable. Requires integration with Kaspersky Threat Intelligence Portal (Premium access).
- **Move to quarantine**. Use this button to move the device on which the file is located to quarantine. This button is only available for hash (MD5 or SHA256) observables.
- **Add prevention rule**. Use this button to add a rule that prevents the file from running. This button is only available for hash (MD5 or SHA256) observables.
- **Delete prevention rule**. Use this button to delete the rule that prevents the file from running. This button is only available for hash (MD5 or SHA256) observables.
- **Terminate process**. Use this button to terminate processes associated with the file. This button is only available for hash (MD5 or SHA256) observables.

- **Similar closed alerts**

In the **Similar closed alerts** section you can view the list of closed alerts that have the same affected artifacts as the current alert. The affected artifacts include observables and affected devices. The similar closed alerts can help you investigate the current alert.

By using the list, you can evaluate the degree of similarity of the current alert and other alerts. The similarity is calculated as follows:

Similarity = M / T * 100

Here, 'M' is a number of artifacts that matched in the current and a similar alert, and 'T' is total number of artifacts in the current alert.

If the similarity is 100%, the current alert has nothing new in comparison with the similar alert. If the similarity is 0%, the current and the similar alert are completely different. Alerts that have a similarity of 0% are not included in the list.

The calculated value is rounded off to the nearest whole number. If similarity is equal to a value between 0% and 1%, the application does not round such a value down to 0%. In this case, the value is displayed as less than 1%.

Clicking an alert ID opens the alert details.

Customizing the similar closed alerts list

You can customize the table by using the following options:

- Filter the alerts by selecting the term for which the alerts have been updated. By default, the list contains the alerts that have been updated for the last 30 days.
- Click the **Columns settings** icon ( ), and then select which columns to display and in which order.
- Click the **Filter** icon ( ), and then select and configure the filters that you want to apply. If you select several filters, they are applied simultaneously by logical AND operator.
- Click a column header, and then select the sorting options. You can sort the alerts in ascending or descending order.

- **Similar incidents**

In the **Similar incidents** section, you can view the list of incidents that have the same affected artifacts as the current alert. The affected artifacts include observables and affected devices. The similar incidents can help you decide if the current alert may be linked to an existing incident.

By using the list, you can evaluate the degree of similarity of the current alert and the incidents. The similarity is calculated as follows:

Similarity = M / T * 100

Here, 'M' is a number of artifacts that matched in the current alert and a similar incident, and 'T' is total number of artifacts in the current alert.

If the similarity is 100%, the current alert has nothing new in comparison with the similar incident. If the similarity is 0%, the current alert and the similar incident are completely different. Incidents that have similarity of 0% are not included in the list.

The calculated value is rounded off to the nearest whole number. If the similarity is equal to a value between 0% and 1%, the application does not round such a value down to 0%. In this case, the value is displayed as less than 1%.

Clicking an incident ID opens the incident details.

Customizing the similar incidents list

You can customize the table by using the following options:

- Filter the incidents by selecting the term for which the incidents have been updated. By default, the list contains the incidents that have been updated for the last 30 days.
- Click the **Columns settings** icon (⚙), and then select which columns to display and in which order.
- Click the **Filter** icon (▽), and then select and configure the filters that you want to apply. If you select several filters, they are applied simultaneously by logical AND operator.
- Click a column header, and then select the sorting options. You can sort the incidents in ascending or descending order.

- **Comments**

In the **Comments** section, you can leave comments related to the alert. For example, you can enter a comment about investigation results or when you change the alert properties, such as the alert assignee or status.

You can edit or remove your own comments. The comments of other users cannot be modified or removed.

To save your comment, press **`Enter`**. To start a new line, press **`Shift+Enter`**. To edit or delete your comment, use the buttons on the top right.

The **Write** permission in the **Alerts and incidents** functional area is required to leave comments.

- **History**

In the **Change log** section, you can track the changes that were made to the alert as a work item:

- Changing alert status
- Changing alert assignee
- Linking alert to an incident
- Unlinking alert from an incident
- Uploading a file to the alert
- Deleting a file from the alert

In the **Response history** section, you can review the log of manual and playbook response actions. The table contains the following columns:

- **Time**. The time when the event occurred.
- **Launched by**. Name of the user who launched the response action.
- **Events**. Description of the event.
- **Response parameters**. Response action parameters that are specified in the response action.

- **Asset**. Number of the assets for which the response action was launched. You can click the link with the number of the assets to view the asset details.
- **Action status**. Execution status of the response action. The following values can be shown in this column:
  - **Awaiting approval**—Response action awaiting approval for launch.
  - **Launched**—Response action is in progress.
  - **Success**—Response action is completed without errors or warnings.
  - **Warning**—Response action is completed with warnings.
  - **Error**—Response action is completed with errors.
  - **Terminated**—Response action is completed because the user interrupted the execution.
  - **Approval time expired**—Response action is completed because the approval time for the launch has expired.
  - **Rejected**—Response action is completed because the user rejected the launch.
- **Playbook**. Name of the playbook in which the response action was launched. You can click the link to view the playbook details.
- **Response action**. Name of the response action that was performed.
- **Asset type.** Type of asset for which the response action was launched. Possible values: **Device** or **User**.
- **Asset tenant**. The tenant that is the owner of the asset for which the response action was launched.

Assigning alerts to analysts

As a work item, an alert can be assigned to a SOC analyst or to a user group for inspection and possible investigation. You can assign alerts to a user group if you want to automatically assign alerts to analysts within the selected group. For details, refer to Configuring the automatic assignment of alerts and incidents.

You can change the assignee of an active alert at any time. You cannot change an assignee of a closed alert.

Alerts can be assigned only to analysts and user groups who have the access right to read and modify alerts and incidents.

To assign one or several alerts:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. Select the check boxes next to the alerts that you want to assign.

   You must select only the alerts detected in the same tenant. Otherwise, the **Assign** button will be disabled.

Alternatively, you can assign an alert to an analyst or to a user group from the alert details. To open the alert details, click the link with the alert ID you need.

3. Click the **Assign** button.

4. In the **Assign to analyst** window that opens, select whether to assign the alerts to a user or a user group.

5. Start typing the name of an analyst or a user group, and then select the assignee from the list.

   If you want to unassign alerts, select **Not assigned**.

   > You can select **Not assigned** for all alerts, except alerts with the **Closed** status.

6. Click the **Assign** button.

Alerts are assigned.

You also can assign an alert by using playbooks.

Changing an alert status

As a work item, an alert has a status that shows the current state of the alert in its life cycle.

> You can change alert statuses for your own alerts or the alerts of other analysts only if you have the access right to read and modify alerts and incidents.

> If the alert status is changed manually, playbooks will not launch automatically. You can launch a playbook for such an alert manually.

An alert can have one of the following statuses:

- **New**

  > When Open Single Management Platform registers a new alert, the alert has the *New* status. You can change the status to *In progress* or *Closed*. When you change the *New* status to *Closed,* and the alert has no assignee, the alert is automatically assigned to you.

- **In progress**

  > This status means that an analyst started working on the alert. You can change the *In progress* status to *New* or *Closed*.

- **Closed**

  > True positive alerts are to be linked to incidents and be investigated within the incidents. When you close an incident, the linked alerts also gain the *Closed* status. You close an unlinked alert only as false positive or a low-priority alert. When you close an alert, you must select a resolution.

The *Closed* status can only be changed to status *New*. If you want to return a closed alert back to active, change its status as follows: *Closed* → *New* → *In progress*.

When you close an alert linked to an incident, the alert is automatically unlinked from the incident. If the alert that you are going to close has no assignee, the alert is automatically assigned to the analyst who closes the alert.

- **In incident**

  Alerts gain this status when they are linked to an incident. You cannot set this status manually. You can only set the *Closed* status to a linked alert. To set the *New* or *In progress* status, you first must unlink the alert from the incident.

To change the status of one or several alerts:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. Do one of the following:
   - Select the check boxes next to the alerts whose status you want to change.
   - Click the link with the ID of the alert whose status you want to change.

     The **Alert details** window opens.
3. Click the **Change status** button.
4. In the **Change status** pane, select the status to set.

   If you select the *Closed* status, you must select a resolution.

   If you change the alert status to *Closed* and this alert contains uncompleted playbooks or response actions, all related playbooks and response actions will be terminated.
5. Click the **Save** button.

The status of the selected alerts is changed.

If an alert is added to the investigation graph, you can also change the alert status through the graph.

You also can change the alert status by using playbooks.

Creating alerts manually

You can create an alert manually from a set of events. You can use this functionality to examine a hypothetical incident that has not been detected automatically.

If the alert is created manually, playbooks will not launch automatically. You can launch a playbook for such an alert manually.

To create an alert manually:

1. In the main menu, go to **Monitoring** → **Threat hunting**.
2. Select the events for which you want to create an alert. The events should belong to the same tenant.
3. Click the **Create alert** button.

   A window shows up that displays the created alert, and you receive a corresponding email notification if notifications templates are configured. The **Severity** field value corresponds to the maximum severity among the selected events.

Manually created alerts have a blank **Rules** value in the **Monitoring** → **Alerts** table.

Linking alerts to incidents

You can link one or multiple alerts to an incident for the following reasons:

- Multiple alerts may be interpreted as indicators of the same issue in an organization's IT infrastructure. If this is the case, the alerts in the incident can be investigated as a single issue. You can link up to 200 alerts to an incident.
- A single alert may be linked to an incident if the alert is defined as true positive.

You can link an alert to an incident if the alert has any status other than *Closed*. When linked to an incident, an alert loses its current status and gains the special status *In incident*. If you link alerts that are currently linked to other incidents, the alerts are unlinked from the current incidents, because an alert can be linked to only one incident.

> Alerts can only be linked to an incident that belongs to the same tenant.

Alerts can be linked to an incident manually or automatically.

Linking alerts manually

To link alerts to an existing or new incident:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. Select the check boxes next to the alerts that you want to link to an incident.
3. If you want to link alerts to an existing incident:

   a. Click the **Link to incident** button.

   b. Select an incident to link the alerts to.

   Alternatively, click an alert to display its details and click the **Link to incident** button in the toolbar at the top.

4. If you want to link alerts to a new incident:

   a. Click the **Create incident** button.

   b. Fill in the properties of the new incident: name, assignee, priority, and description.

   Alternatively, click an alert to display its details and click the **Create incident** button in the toolbar at the top.

5. Click the **Save** button.

The selected alerts are linked to an existing or new incident.

Linking alerts automatically

If you want alerts to automatically link to an incident, you have to configure segmentation rules.

Unlinking alerts from incidents

You might need to unlink an alert from an incident, for example, if the alert analysis and investigation showed that the alert is not connected to other alerts in the incident. When you unlink an alert from an incident, Open Single Management Platform performs the following actions:

- Refreshes all of the data related to the incident, to reflect that the alert no longer belongs to the incident. For example, you can view the changes in the incident details.
- Resets the status of the unlinked alerts to *New*.

To unlink an alert from an incident:

1. Open the alert details.
2. Click the **Unlink from incident** button in the toolbar at the top.

   The **Unlink alerts** window opens.
3. If you want to change the assignee, select **Assign the alerts to**, and then specify the new assignee.
4. If you want to add a comment, specify it in the **Comment** section. The comment you specify will be displayed in the **Details** column in the **History** section.

The selected alerts are unlinked from the incident.

Linking events to alerts

If during the investigation you found an event that is related to the alert being investigated, you can link this event to the alert manually.

You can link an event to an alert that has any status other than **Closed**.

To link an event to an alert:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. In the list of alerts, click the link with the ID of the alert to which you want to link the event.

   The **Alert details** window opens.
3. Go to the **Details** section, and then click the **Find in Threat hunting** button.

   The **Threat hunting** section opens. By default, the event table contains events related to the selected alert.

   > The event table contains only events related to tenants that you have access to.
4. In the upper part of the window, open the first drop-down list, and then select **Storage**.
5. Open the third drop-down list, and then specify the time range.

   You can select predefined ranges relative to the current date and time, specify a custom range by using the **Range start** and **Range end** fields, or by selecting dates in the calendar.
6. Click the **Run query** button.
7. In the updated list of events, select an event that you want to link to the alert, and then click **Link to alert**.

The selected events are linked to the alert.

Unlinking events from alerts

You might need to unlink an event from an alert, for example, if the alert analysis and investigation showed that the event is not connected to the alert.

To unlink an event from an alert:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. In the list of alerts, click the link with the ID of the alert from which you want to unlink the event.

   The **Alert details** window opens.
3. In the **Details** section, select the events that you want to unlink, and then click the **Unlink from alert** button.

The selected event are unlinked from the alert.

Working with alerts on the investigation graph

On the investigation graph, you can perform the following actions with alerts:

- Add an alert to the graph.
- Hide an alert from the graph.
- View an alert details by selecting the corresponding item from the context menu of the alert node.

- Change an alert status.
- View events related to an alert.
- View assets related to an alert.
- View observables related to an alert.

Adding alerts to the investigation graph

You can add an alert to the investigation graph in one of the of the following ways:

- From the general table of alerts that opens when you click the **Add alert** button on the investigation graph. You have to select the check boxes next to the alerts that you want to be displayed on the investigation graph, and then click the **Show on graph** button.
- From the table of similar alerts.

To add an alert to the investigation graph from the table of similar alerts:

1. Do one of the following:
   - If on the investigation graph you have an asset, observable, or segmentation rule, click its node, and then in the context menu, click **Find similar alerts**.
   - If on the investigation graph you have an event, click its node, and then in the context menu, click **View details**. In the window that opens, click the **Show on graph** button.
   - If on the investigation graph you have an alert, click its node, and in the context menu, click **Events**. In the table of events, click the event whose details you want to open. If the event details contain an observable, asset, or segmentation rule, click the link in the corresponding field, and then in the context menu, click **Find similar alerts**.
   - On the investigation graph, click the **Threat hunting** button, and then in the general table of events, click the event whose details you want to open. If the event details contain an observable, asset, or segmentation rule, click the link in the corresponding field, and then in the context menu, click **Find similar alerts**.

   The table of similar alerts is displayed.
2. Select the check boxes next to the alerts that you want to be displayed on the investigation graph, and then click the **Show on graph** button.

The selected alerts are added to the investigation graph.

Hiding alerts from the investigation graph

You can hide an alert from the investigation graph in one of the following ways:

- By clicking the alert node and selecting **Hide** in the context menu.
- Through the table of alerts.

To hide an alert from the graph through the table of alerts:

1. Do one of the following:

   - In the toolbar at the top of the investigation graph, click the **Add alert** button.

   - If you have observables, assets, or events nodes displayed on the graph, click the node for which you want to add an alert, and then in the context menu, select **Find similar alerts**.

   The table of alerts is displayed.

2. Select the check boxes next to the alerts that you want to hide from the investigation graph, and then click the **Show on graph** button.

   The selected alerts and their links will be hidden from the investigation graph. The related nodes remain on the investigation graph.

Changing an alert status

To change an alert status:

1. Click the alert node, and in the context menu, select **Change status**.

2. In the **Change status** pane that opens, select the status, and then click **Save**.

   If you select the *Closed* status, you must select a resolution.

   The status of the selected alerts is changed.

Viewing the events related to an alert

To view events related to an alert, do one of the following:

- Click the digit next to the alert node for which you want to display the events. The digit shows the number of events related to the alert.

- Click the alert node for which you want to display the events, and then in the context menu, click **Events**.

If you want to add the events from the table to the investigation graph, select the check boxes next to the events, and then click the **Show on graph** button.

If you want to hide the events from the investigation graph, select the check boxes next to the events, and then click the **Hide on graph** button.

Viewing assets related to an alert

To view assets related to an alert, click the alert node.

In the context menu, the digits next to the **Devices** and **Users** items show the number of devices and users related to the alert.

If you want to add devices or users to the investigation graph, click the corresponding menu item.

Viewing observables related to an alert

To view observables related to an alert, click the alert node, and in the context menu, click **Events**.

In the menu that opens, the digits next to the items show the number of observables relate related to the alert.

If you want to add an observable (for example, **Hash**, **Domain**, **IP address**) to the investigation graph, click the corresponding menu item.

Adding an IOA rule to exclusions

If you do not want to generate alerts for events generated as part of asset activity that is normal for your organization, you can add an IOA rule to exclusions.

To add an IOA rule to exclusions from the alert table:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. Click the filter icon.
3. On the **Filters** tab, specify and apply the **Detected** filter criterion for the desired IOA rule.
4. In the alert table, click the ID of an alert.

   The alert details are displayed.
5. In the **Summary** section, click the IOA rule name.
6. In the IOA rule panel, click **Add to exclusions**.

   The **Add to exclusions** panel is invoked.
7. In the **Add to exclusions** panel, select a policy or profile to add the IOA rule to exclusions. Click **Next**.
8. Specify the exclusion rule properties and click **Add**.

   The IOA rule is added to exclusions.

To view and edit the list of IOA exclusions for a policy:

1. In the main menu, go to **Policies**.
2. Click the desired policy.
3. Select the **Application settings** tab. Expand the **General settings** menu on the left and select **Telemetry Settings**.

   In the **EDR Expert (on-premise) telemetry** tab, the **General exclusions** tab shows IOA exclusions.

   - To disable an exclusion rule, turn off its **Status** toggle switch.
   - To delete an exclusion rule, select the check box next to the exclusion rule and click **Delete**.
   - To view exclusion rule details, click the exclusion rule name.

Managing aggregation rules

This section contains general information about aggregation rules and gives instructions on how to create, edit, duplicate, copy to another tenant, and delete aggregation rules.

> To manage aggregation rules, you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator.

Aggregation rules allow you to combine repetitive events of other Kaspersky solutions into Open Single Management Platform alerts. The alerts are either linked to the existing incidents, or form new incidents.

You can create and manage aggregation rules in the **Aggregation rules** section of tenant properties. The aggregation rules are displayed in a list. The higher the rule is in the list of aggregation rules, the higher its priority. If you want to change the aggregation rule priority, you have to drag and drop the rule by clicking the drag (⠿) icon.

Also, the list contains the predefined aggregation rule created by Kaspersky experts. This rule combines the events for which the same correlation rule was triggered during the default aggregation interval (30 seconds).

You cannot delete the predefined aggregation rule, but you can edit it. By default, the rule is enabled and always displayed in the aggregation rules table, with the **Kaspersky Lab** value in the **Created by** column.

The aggregation rules are processed sequentially according to the specified priority. When the condition set in the **Trigger** section is met, the aggregation rule is triggered, and the search for other suitable aggregation rules continues. The search stops only when all the rules in the list are processed.

Limitations related to detection technologies

Depending on detection technology, there are some limitations on alert aggregation:

- YARA, Sandbox and Anti-Malware detection on file:
  - You can aggregate detections by these technologies on the same file.
  - You cannot add IOA detections or the results of IOC or YARA scans.
- IOA detection:
  - You can aggregate into one alert detections triggered by one IOA rule or different IOA rules.
  - You cannot add detections by all other technologies.
- IOC scans:
  - You can aggregate the results of IOC scans (for example, detections by the same IOC rule, different IOC rules, the same or different hosts).
  - You cannot add detections by all other technologies.

- YARA scans:
  - You can aggregate the results of YARA scans (for example, detections by the same YARA rule, different YARA rules, the same or different hosts).
  - You cannot add detections by all other technologies.

Creating aggregation rules

Aggregation rules help you to combine repetitive events of other Kaspersky solutions into Open Single Management Platform alerts.

> To create aggregation rules, you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator.

To create an aggregation rule:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Select the **Settings** tab, and then click **Aggregation rules**.

   The list of aggregation rules opens.

4. Click the **Create** button.

   The **Create aggregation rule** window opens.

5. In the **Status** field, enable the aggregation rule you want to create.

   By default, the rule is disabled.

6. In the **Rule name** field, specify the aggregation rule name.

   The rule name must be unique.

7. In the **Max events in alert** field, specify the maximum number of events combined in an alert.

   The value must be between 1 and 100. The default value is 100.

8. In the **Aggregation interval** field, specify the searching interval in minutes or seconds.

   The maximum value is 1500 seconds. The default value is 30.

9. If necessary, in the **Description** field, specify a description of the aggregation rule.

10. In the **Alert name** section, specify the jq expression that defines the template that will be used to create the name of the new alert. This expression is also used to combine events into an alert.

    By default, the following expression is specified: `([.Rules[]?.Name // empty] | sort | join(","))`

    When creating a new alert, the name is generated from the values of the fields specified in the template.

    If necessary, you can copy the jq expression for the **Alert name** section from another aggregation rule. To do that, click the **Copy from another rule** button.

Example of the jq expression for the Alert name section

```
if any(.OriginalEvents[]?.BaseEvents[]?.DestinationUserName // empty; startswith("adm_")) then "Brute force admin" else "Brute force not admin" end
```

11. In the **Trigger** section, specify the jq expression that defines the condition for creating an alert from an event.

    If necessary, you can copy the jq expression for the **Trigger** section from another aggregation rule. To do that, click the **Copy from another rule** button.

    Example of the jq expression for the Trigger section

    ```
    .AggregationID | startswith("bruteforce_DestinationUserName")
    ```

12. Click the **Create** button.

    The aggregation rule is created.

By default, the aggregation rule is created with the lowest priority. To change the aggregation rule priority, in the list of aggregation rules, drag and drop the rule by clicking its name. The higher the rule is in the list of aggregation rules, the higher its priority.

Editing aggregation rules

If necessary, you can edit the aggregation rule.

To edit the aggregation rule:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Select the **Settings** tab, and then click **Aggregation rules**.

4. In the list of aggregation rules that opens, click the name of the aggregation rule that you want to edit.

5. In the **Edit aggregation rule** window that opens, edit the aggregation rule properties. For more details on the rule properties that you can edit, refer to Creating aggregation rules.

6. Click the **Apply** button.

7. If you want to change the aggregation rule priority, in the list of aggregation rules, drag and drop the rule by clicking its name.

   The higher the rule is in the list of aggregation rules, the higher its priority.

The properties of the aggregation rule are modified and saved.

Copying aggregation rules to another tenant

You can copy an existing aggregation rule to another tenant.

To copy aggregation rules:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Select the **Settings** tab, and then click **Aggregation rules**.

4. In the list of aggregation rules, select the rules that you want to copy, and then click **Copy to tenant**.

5. Select one or multiple target tenants, and then click **Copy**.

6. If the target tenant contains an aggregation rule with an identical name, the **Overwrite or rename aggregation rules?** window appears:

   - If you want to delete the previously created rule for the target tenant and replace it with the rule that you want to copy, click **Overwrite**.

   - If you want to preserve the previously created rule and copy the specified rule with (copy) appended to its title, click **Copy and rename**.

   When you click the close (×) button, the window is closed, and the aggregation rules whose names are identical to the ones in the target tenants are not copied. However, if at step 4 you also selected the aggregation rules that are absent in the target tenants, such rules are copied.

The aggregation rules are copied to another tenant or tenants.

Deleting aggregation rules

You can permanently delete one or multiple aggregation rules at once.

You cannot delete the predefined aggregation rule.

To delete aggregation rules:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Select the **Settings** tab, and then click **Aggregation rules**.

4. In the list of aggregation rules, select one or multiple aggregation rules that you want to delete, and then click **Delete**.

5. In the confirmation dialog box, click **Delete**.

The aggregation rule is deleted.

Working with incidents

This section contains general information about incidents, their properties, typical life cycle, and connection with alerts. This section also gives instructions on how to create incidents, analyze the incident table, change incident properties according to the current state in the life cycle, and merge incidents.

The **Incidents** section is displayed in the main menu if the following conditions are met:

- You have a license key for Kaspersky Next XDR Expert.
- You are connected to the root Administration Server in OSMP Console.
- You have one of the following XDR roles: Main administrator, Tenant administrator, Junior analyst, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Auditor, Interaction with child incidents.

About incidents

An *incident* is a container of alerts that normally indicates a true positive issue in the organization's IT infrastructure. An incident may contain a single or several alerts. By using incidents, analysts can investigate multiple alerts as a single issue.

You can create incidents manually or enable the rules for automatic creation of incidents. After an incident is created, you can link alerts to the incident. You can link no more than 200 alerts to an incident.

After creation, Open Single Management Platform adds incidents to the incident table as work items that are to be processed by analysts.

Incidents can be assigned only to analysts who have the access right to read and modify alerts and incidents.

You can manage incidents as work items by using the following incident properties:

- **Incident status**

  Possible values: *New*, *In progress*, *On hold*, or *Closed*.

  The incident status shows the current state of the incident in its life cycle. You can change the status as you like, with the following exceptions:

  - Status *New* cannot be changed to *On hold*.
  - Status *Closed* can only be changed to *New*.

- **Incident severity**

  Possible values: **Low**, **Medium**, **High**, or **Critical**.

  The incident severity shows the impact this incident may have on computer security or corporate LAN security, based on Kaspersky experience. An incident's severity corresponds to the highest severity of the linked alerts and cannot be changed manually.

- **Incident priority**

  Possible values: **Low**, **Medium**, **High**, or **Critical**.

  Incident priority defines the order in which the incidents must be investigated by analysts. Incidents with the **Critical** priority are the most urgent ones and must be investigated first. You can change the incident priority manually.

- **Incident assignee**

  This is an incident owner, the analyst who is responsible for the incident investigation and process. You can change an incident assignee at any time if the **Status** parameter is not set to **Closed**.

Two or more incidents may be interpreted as indicators of the same issue in an organization's IT infrastructure. If this is the case, you can merge the incidents to investigate them as a single issue.

Each incident has *incident details* that provide all of the information related to the incident. You can use this information to investigate the incident or merge incidents.

For each incident, you can create child incidents. Child incidents allow you to investigate and respond to incidents across different tenants. You can also create a child incident of another child incident. A parent incident can have no more than 200 child incidents.

Incident data model

The structure of an incident is represented by fields that contain values (see the table below). Some fields are objects or arrays of objects with their own set of fields (for example, the `Assignee` and `Alerts` fields).

Incident

Field	Value type	Is required	Description
`InternalID`	String	Yes	Internal incident ID, in the UUID format.
`ID`	Integer	Yes	Short internal incident ID.
`TenantID`	String	Yes	ID of the tenant that the incident is associated with, in the UUID format.
`IncidentType`	`IncidentType` object	Yes	Incident type.
`Name`	String	Yes	Incident name.
`WorkflowName`	String	Yes	Name of the incident workflow.
`WorkflowUUID`	String	Yes	Unique identifier of the incident workflow, in the UUID format.
`Description`	String	No	Incident description.

`CreatedAt`	String	Yes	Date and time of the incident creation, in the RFC 3339 format.
`UpdatedAt`	String	Yes	Date and time of the last incident change, in the RFC 3339 format.
`StatusChangedAt`	String	No	Date and time of the incident status change, in the RFC 3339 format.
`Severity`	String	No	Severity of the incident. Possible values: • `critical` • `high` • `medium` • `low`
`Priority`	String	Yes	Priority of the incident. Possible values: • `critical` • `high` • `medium` • `low`
`Assignee`	`Assignee` object	No	Operator to whom the incident is assigned.
`FirstEventTime`	String	No	Date and time of the first telemetry event of the alert related to the incident, in the RFC 3339 format.
`LastEventTime`	String	No	Date and time of the last telemetry event of the alert related to the incident, in the RFC 3339 format.
`Status`	String	Yes	Incident status. Possible values: • `open` • `inProgress` • `hold` • `closed`
`StatusUUID`	String	Yes	Incident status ID, in the UUID format.
`StatusResolution`	String	No	Resolution of the incident status. Possible values: • `truePositive` • `falsePositive` • `lowPriority` • `merged`
`DetectSources`	Array of strings	No	Components that detect and generate the incident.
`DetectionTechnologies`	Array of strings	No	Triggered detection technology.
`Alerts`	Array of `Alert` objects	No	Alerts included in the incident.
`AdditionalData`	Object	No	Additional information about the alert, in the JSON format. This information can be filled in by a user or a playbook.
`ExternalRef`	String	Yes	Link to an entity in an external system (for example, a link to a Jira ticket).
`SignOfCreation`	String	Yes	Method of creating an incident.
`Attachments`	Array of `UnkeyedAttachment` objects	No	Attachments related to the incident.

IncidentType

Field	Value type	Is required	Description
`ID`	String	Yes	Incident type ID, in the UUID format.
`Name`	String	Yes	Name of the incident type.
`Description`	String	Yes	Description of the incident type.

Assignee

Field	Value type	Is required	Description
`ID`	String	Yes	User account ID of the operator to whom the incident is assigned.
`Name`	String	Yes	Name of the operator to whom the incident is assigned.

UnkeyedAttachment

Field	Value type	Is required	Description
`AttachmentID`	String	Yes	Attachment ID, in the UUID format.
`Name`	String	Yes	Attachment name.
`CreatedAt`	String	Yes	Date and time of the attachment creation, in the UTC format.
`UpdatedAt`	String	Yes	Date and time of the last attachment change, in the UTC format.
`CreatedBy`	String	Yes	Indicator that the affected asset (a device or an account) is a victim.
`Size`	Integer	Yes	Attachment size, specified in bytes.
`Status`	String	Yes	Attachment status that indicates whether the attachment upload is in progress, completed, or aborted with an error. Possible values: • `completed` • `error` • `uploading`
`Description`	String	No	Attachment description.
`StatusCode`	String	No	Text of the status that is displayed to a user (for example, an error message that is displayed when the attachment upload fails).

Creating incidents

You can create incidents manually or enable the rules for automatic creation of incidents. This topic describes how to create incidents manually.

> To be able to create incidents, you must have the access right to read and modify alerts and incidents.

> If the incident is created manually, playbooks will not launch automatically. You can launch a playbook for such an incident manually.

You can create incidents by using the incident table or the alert table.

Creating incidents by using the incident table

To create an incident:

1. In the main menu, go to **Monitoring** → **Incidents**. Click the **Create incident** button.

2. On the **General settings** step, specify the following settings:

   - Incident name

   - **Tenant**

     A tenant that the incident is associated with. Alerts can only be attached to an incident that belongs to the same tenant. You cannot change the incident's tenant later.

   - **Assignee**

     This is an incident owner, the analyst who is responsible for the incident investigation and process. You can change an incident assignee at any time if the **Status** parameter is not set to **Closed**.

   - **Priority**

     Possible values: **Low**, **Medium**, **High**, or **Critical**.

     Incident priority defines the order in which the incidents must be investigated by analysts. Incidents with the **Critical** priority are the most urgent ones and must be investigated first. You can change the incident priority manually.

   - **Description**

     In this field, you can leave a description of the incident. For example, you can describe the issue or provide investigation results of the linked alerts. The description is added to the **Description** section of the incident details.

     This field is optional.

3. Click **OK**.

   The incident is created.

Creating incidents by using the alert table

You create an incident by selecting the alerts to link to the new incident. Refer to linking alerts to incidents.

Creating child incidents

Child incidents allow you to monitor the investigation progress and respond to incidents across different tenants. You can also create a child incident of another child incident. A parent incident can have up to 200 child incidents.

You must have one of the following XDR roles: Main administrator, Tenant administrator, Junior analyst, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Interaction with child incidents.

To create a child incident:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident.

   The **Incident details** window opens.
3. In the **Child incidents** section, click the **Create** button.

   The **Create child incident** window opens.
4. If necessary, in the **Name** field, change the name of the child incident.

   By default, the child incident name includes the [Child] tag and the parent incident name.
5. In the **Tenant** drop-down list, select the tenant to assign the child incident to.
6. In the **Analyst** drop-down list, start typing the analyst's name, and then select the analyst from the list.

   You can also select the **Not assigned** option.
7. If necessary, in the **Priority** drop-down list, change the incident priority.

   By default, the child incident inherits the priority from the parent incident.
8. If necessary, in the **Description** field, specify a description of the child incident.
9. If you want to copy all the information from the parent incident, select the **Copy information from the parent incident** check box.

   If this check box is selected, all information, including observables, assets, and linked alerts, will be copied from the parent incident. The copied alerts will be assigned to the tenant of the child incident.

   If this check box is cleared, the child incident will not contain information about assets, observables, and linked alerts from the parent incident.
10. Click the **Create** button.

The child incident is created and linked to the parent incident. You can find the child incident in the **Child incidents** section of the **Incident details** window.

If a child incident is assigned to a tenant that does not have the incident type that the parent incident belongs to, the child incident is assigned the default incident type and the default incident workflow. If necessary, you can change the incident type.

Viewing the incident table

The incident table is displayed in the **Monitoring** → **Incidents** section and provides an overview of all created incidents. By default, the table displays the incidents related to all of the tenants to which you have access rights. If necessary, you can configure the incident table.

To configure the incident table, do any of the following:

- Apply tenant filter:

  a. Click the link next to the **Tenant filter** setting.

  The tenant filter opens.

  b. Select the check boxes next to the required tenants.

- Filter the data of the alerts table:

  a. Click the filter ( ) icon.

  b. On the **Filters** tab, specify and apply the filter criterion in the invoked menu.

- If you want to hide or display a column, click the settings ( ) icon, and then select the necessary column.

The incident table is configured and displays the data you need.

The incident table has the following columns:

- **Created**. Date and time when the incident was created.
- **Threat duration**. Time between the earliest and the most recent events among all of the alerts linked to the incident.
- **Updated**. Date and time of the last change, from the incident history.
- **Incident ID**. A unique identifier of an incident.
- **Status**. Current status of the incident.
- **Time to triage**. The time limit to accept the incident, in days, hours and minutes.
- **Time spent to respond**. The time limit for initial response to the incident, in days, hours and minutes.
- **Time to resolve**. The time limit to resolve the incident, in days, hours and minutes.
- **Status changed**. The date and time when the incident status has been changed.
- **Severity.** Severity of the incident.
- **Priority**. Priority of the incident.
- **Linked alerts**. How many alerts are included in the incident.

- **Name**. A name of an incident.
- **Rules**. The rules that were triggered to create the incident.
- **Affected assets**. Devices and users that were affected by the incident. If the number of assets affected by or involved in the incident is greater than or equal to three, the number of affected devices is displayed.
- **Tenant**. The name of the tenant in which the incident was detected.
- **Assignee**. Current assignee of the incident.
- **Has parent incident**. Displays whether the incident has a parent incident. If the **Yes** value is displayed, then the incident is a child incident.
- **Status of child incidents**. Resulting status of first-level child incidents. Possible values: **Closed**, **In progress**, **No child incidents**.
- **Creation method**. How the incident was created—manually or automatically.
- **Observables**. Number of the detection artifacts, for example, IP addresses or MD5 hashes of files. If the number of observables is greater than or equal to three, the number of observables is displayed.

If necessary, you can export information about all incidents displayed in the incident table to a JSON file.

Viewing incident details

Incident details are a page in the interface that contains all of the information related to the incident, including the incident properties.

To view incident details:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident.

   The window with incident details is displayed.

If necessary, you can refresh the information in the incident details by clicking the refresh (⟳) icon next to the incident name.

The toolbar in the upper part of the incident details allows you to perform the following actions:

- Edit the **Name**, **Description** and **External reference** field values
- View the incident on the investigation graph
- Change the incident status
- Change the incident priority
- Assign the incident to an analyst
- Select a playbook

- Link alerts to the incident
- Merge the incident with other incidents

Incident details contain the following sections:

- **Summary**

  The summary section contains the following incident properties:

  - **Type**. Incident type.
  - **Analyst**. Current assignee of the incident.
  - **Creation method**. How the incident was created—manually or automatically.
  - **Name**. Name specified at the incident creation. You can click the **Edit** button at the top to change the incident name.
  - **Tenant**. Name of the tenant in which the incident was detected.
  - **Assets**. Number of users and devices that were affected by the incident.
  - **Registered**. Date and time when the incident was created.
  - **Updated**. Date and time of the last change from the incident history.
  - **First event**. A date and time of the first event related to the incident. This is the earliest event in the Details section of the alert details among all of the alerts linked to the incident.
  - **Last event.** A date and time of the most recent event related to the incident. This is the most recent event in the Details section of the alert details among all of the alerts linked to the incident.
  - **Description.** Incident description. You can click the **Edit** button at the top to specify the description.
  - **External reference.** Link to an entity in an external system. You can click the **Edit** button at the top to specify the external reference.
  - **Priority**. Possible values: **Low**, **Medium**, **High**, or **Critical**. Incident priority defines the order in which the incidents can be investigated. Incidents with the **Critical** priority are the most urgent ones and can be investigated first. You can change the priority by clicking the current priority value.
  - **Severity**. Possible values: **Low**, **Medium**, **High**, or **Critical**. Incident severity shows the impact this incident may have on computer security or corporate LAN security based on Kaspersky experience.
  - **Time to triage**—The time limit to accept the incident, in days, hours and minutes.
  - **Time to respond**—The time limit for initial response to the incident, in days, hours and minutes.
  - **Time to resolve**—The time limit to resolve the incident, in days, hours and minutes.

- **Correlation rules**. The rules that were triggered to detect the linked alerts. By clicking the ellipsis icon next to the rule name, you can open the shortcut menu. Use this menu to learn more details about the rule, find alerts or incidents that were detected by the same rule, or search the rule-triggering events in **Threat hunting** for the period between the first and the last event of the incident.

  When you click **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

- **IOA rules**. The IOA rules that were triggered to detect a threat and create the related alert. You can open the context menu by clicking an IOA rule name. By using the context menu, you can do the following:

  - View the detailed information about that IOA rule by clicking the **Open properties** button.
  - View the related events by clicking the **Go to Threat hunting** button.
  - Find similar incidents.
  - View the detailed information about an IOA rule in Kaspersky TIP by clicking the **Find in Kaspersky TIP** button.
  - Copy the IOA rule name.

- **MITRE tactic**. A tactic or several tactics detected in the alerts linked to the incident. The tactics are defined in the MITRE ATT&CK knowledge base.
- **MITRE technique**. A technique or several techniques detected in the alerts linked to the incident. The techniques are defined in the MITRE ATT&CK knowledge base.
- **Additional data**. Additional information on the incident. You can edit a value in this field only by using a playbook. The field is displayed if you added a value.

- **Details**

In the **Details** section, you can track the telemetry events related to the incident.

To view the events related to the incident, click the **Find in Threat hunting** button. The opened table displays alert events related to the incident.

The toolbar of the event table allows you to perform the following actions:

- **Download events**. Click this button to download information about related events into a TSV file.
- **Unlink from incident**. You can select an event or several events in the table, and then click this button to unlink the selected events from the alert related to the incident.

You can go back to the incident details by clicking **Incident investigation** or by clicking the back button in your browser.

- **Similar incidents**

In the **Similar incidents** section, you can view the list of incidents that have the same affected artifacts as the current incident. The affected artifacts include both observables and affected devices of the alerts linked to an incident. The list contains incidents in any status.

By using the list, you can evaluate the degree of similarity of the current incident and other incidents. The similarity is calculated as follows:

Similarity = M / T * 100

Here, M is a number of artifacts that matched in the current and a similar incident, and T is total number of artifacts in the current incident.

If the similarity is 100%, the current incident has nothing new in comparison with the similar incident. If the similarity is 0%, the current and the similar incident are completely different. Incidents that have similarity of 0% are not included in the list.

The calculated value is rounded off to the nearest whole number. If similarity is equal to a value between 0% and 1%, the application does not round such value down to 0%. In this case, the value is displayed as less than 1%.

Clicking an incident ID opens the incident details.

Customizing the similar incidents list

You can customize the table by using the following options:

- Filter the incidents by selecting the term for which the incidents have been updated. By default, the list contains the incidents that have been updated for the last 30 days.
- Click the **Columns settings** icon ( ), and then select which columns to display and in which order.
- Click the **Filter** icon ( ), and then select and configure the filters that you want to apply. If you select several filters, they are applied simultaneously by logical AND operator.
- Click a column header, and then select the sorting options. You can sort the incidents in ascending or descending order.

- **Alerts**

In the **Alerts** section, you can view the list of the alerts linked to the current incident.

By clicking an alert ID, you can open the alert details. You can also use the toolbar buttons to unlink alerts from the incident.

- **Assets**

In the **Assets** section, you can do the following:

- View the devices and users affected by or involved in the alert.
- Perform response actions. To do this, select the check box next to the asset, and then in the **Select response action** drop-down list, select the required response action.

- Export information about affected assets to a CSV file by clicking the **Export to CSV** button.
- Connect to the asset by using the remote terminal. To do this, click the **Open remote terminal** button.

The asset table contains the following columns:

- **Asset type**

  Possible values: device or user.

- **Asset name**
- **Asset ID**
- **Authorization status**

  This parameter is only applied to device asset type. A device authorization status is defined by KICS for Networks. You can change the authorization status by applying the corresponding response action to a device.

- **Has signs of**

  Possible values: attacker or victim.

- **Administration Server**

  The Administration Server that manages the device.

- **Categories**

  Asset categories which include the asset.

- **Count**

  The number of devices and users affected by or involved in the incident that have the same ID, type, and name.

- **Administration Group**

  The administration group to which the device belongs.

By clicking a user name or a device name, you can:

- View the user or the device properties.

  To view the list of alerts or incidents related to the device, in the device properties window, click the **Show related** button and select either **Alerts** or **Incidents**, depending on what you want to view.

- Copy the user name or the device name in the clipboard.
- Search the user name or the device ID in other alerts.
- Search the user name or the device ID in other incidents.

- Search the user name or the device ID in **Threat hunting** for the period between the first and the last event of the incident.

  When after clicking a user name or a device name you select **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

By clicking a user ID or a device ID, you can:

- View the user or the device properties.

  To view the list of alerts or incidents related to the device, in the device properties window, click the **Show related** button and select either **Alerts** or **Incidents**, depending on what you want to view.

- Copy the user ID or the device ID in the clipboard.
- Search the user ID or the device ID in other alerts.
- Search the user ID or the device ID in other incidents.
- Search the user ID or the device ID in **Threat hunting** for the period between the first and the last event of the incident.

  When after clicking a user ID or a device ID you select **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

- **Child incidents**

  In the **Child incidents** section, you can view the list of incidents that are child incidents of the incident you are viewing.

- **Files**

  The **Files** section contains files obtained through response actions or manually added to this incident. You can upload, download, delete files related to the incident, or edit the files' description.

  You can upload files of any extension. Duplicate file names are allowed. The maximum number of files that you can attach to the incident and the maximum total file size you can specify in the configuration file.

  Default limitations:

  - The number of files cannot exceed 100 per incident.
  - Total file size cannot exceed 25 MB per incident.

  To upload files, click the **Upload** button and select one or multiple files. If you attempt to upload files exceeding the limitations, the **Uploading files** panel displaying a warning message will open. In this panel, you can remove files from the upload queue until the warning message disappears and click the **Upload** button to upload files. If you click the **Upload** button ignoring the warning message, upload will fail and the file list will include files that could not be uploaded with a warning icon next to the file names.

Click a file to open the **Edit file** panel that displays file details. In this panel, you can edit file description.

Use check boxes to select a file or multiple files. Select one or more files and click the **Download** button to download them as an archive in the ZIP format. The archive is protected with the password *infected*. Select a file or multiple files and click the **Delete** button to delete the selected files.

The **Write** permission in the **Alerts and incidents** functional area is required to upload and delete files and edit file descriptions. The **Read** permission in the **Alerts and incidents** functional area is required to download files.

- **Observables**

In the **Observables** section, you can view the observables that relate to the alerts linked to the current incident.

You also can export information about the related observables to a CSV file by clicking the **Export to CSV** button.

The observables may include:

- MD5 hash
- URL
- Domain name
- SHA256
- UserName
- HostName
- IP address

The following private IP addresses are not included in the observables that relate to the alerts linked to the current incident:
From 10.0.0.0 to 10.255.255.255 with the subnet mask 255.0.0.0 or /8
From 172.16.0.0 to 172.31.255.255 with the subnet mask 255.240.0.0 or /12
From 192.168.0.0 to 192.168.255.255 with the subnet mask 255.255.0.0 or /16
From 100.64.0.0 to 100.127.255.255 with the subnet mask 255.192.0.0 or /10

To include private IP addresses in the observables, run the following commands on the administrator host:

```
k0s kubectl -n irp set env deploy/transformer -c transformer
OSMP_TRANSFORMER_FILTER_PRIVATE_IPS_DISABLED=true
```

```
k0s kubectl -n irp set env deploy/incident -c incident
OSMP_FILTER_PRIVATE_IPS_DISABLED=true
```

By clicking a link in the **Value** column, you can:

- Search the observable value in **Threat hunting** for the period between the first and the last event of the incident.

  When after clicking a link in the **Value** column you select **Go to Threat hunting** in the menu, the **Threat hunting** section opens in the same tab. If you want to open the **Threat hunting** section in a new browser tab, click and hold the `Ctrl` key, and then click **Go to Threat hunting** in the menu.

- Search the observable value in other alerts.
- Search the observable value in other incidents.
- Copy the observable value in the clipboard.

The toolbar of this section contains the following buttons:

- **Request status from Kaspersky TIP**. Use this button to obtain detailed information about the selected observable from Kaspersky Threat Intelligence Portal (Kaspersky TIP). As a result, the information is updated in the **Status update** column. Requires integration with Kaspersky Threat Intelligence Portal (Premium access).
- **Enrich data from Kaspersky TIP**. Use this button to obtain detailed information about all of the listed observables from Kaspersky TIP. As a result, the information is updated in the **Enrichment** column. Use a link in the **Enrichment** column to open the obtained enrichment details about an observable. Requires integration with Kaspersky Threat Intelligence Portal (Premium access).
- **Move to quarantine**. Use this button to move the device on which the file is located to quarantine. This button is only available for hash (MD5 or SHA256) observables.
- **Add prevention rule**. Use this button to add a rule that prevents the file from running. This button is only available for hash (MD5 or SHA256) observables.
- **Delete prevention rule**. Use this button to delete the rule that prevents the file from running. This button is only available for hash (MD5 or SHA256) observables.
- **Terminate process**. Use this button to terminate processes associated with the file. This button is only available for hash (MD5 or SHA256) observables.

- **History**

  In the **Incident log** section, you can track the changes that were made to the incident as a work item:

  - Changing incident status
  - Changing incident assignee
  - Linking an alert to the incident
  - Unlinking an alert from the incident
  - Creating a child incident
  - Merging the incident with other incidents

- Uploading a file to the incident
- Deleting a file from the incident

In the **Response history** section, you can see the log of manual and playbook response actions. The table contains the following columns:

- **Time**. The time when the event occurred.
- **Launched by**. Name of the user who launched the response action.
- **Events**. Description of the event.
- **Response parameters**. Response action parameters that are specified in the response action.
- **Asset**. Number of the assets for which the response action was launched. You can click the link with the number of the assets to view the asset details.
- **Action status**. Execution status of the response action. The following values can be shown in this column:
  - **Awaiting approval**—Response action awaiting approval for launch.
  - **Launched**—Response action is in progress.
  - **Success**—Response action is completed without errors or warnings.
  - **Warning**—Response action is completed with warnings.
  - **Error**—Response action is completed with errors.
  - **Terminated**—Response action is completed because the user interrupted the execution.
  - **Approval time expired**—Response action is completed because the approval time for the launch has expired.
  - **Rejected**—Response action is completed because the user rejected the launch.
- **Playbook**. Name of the playbook in which the response action was launched. You can click the link to view the playbook details.
- **Response action**. Name of the response action that was performed.
- **Asset type.** Type of asset for which the response action was launched. Possible values: **Device** or **User**.
- **Asset tenant**. The tenant that is the owner of the asset for which the response action was launched.

- **Effort log**

In the **Effort log** section, you can view the effort that analysts spent on incident-related work.

In this section, you manage your effort. You cannot modify or delete effort added by other analysts.

By clicking a link in the **Data** column, you can view and edit your effort.

- **Comments**

  In the **Comments** section, you can leave comments related to the incident. For example, you can enter a comment about investigation results or when you change the incident properties, such as the incident assignee or status.

  You can edit or remove your own comments. The comments of other users cannot be modified or removed.

  To save your comment, press **`Enter`**. To start a new line, press **`Shift+Enter`**. To edit or delete your comment, use the buttons on the top right.

  The **Write** permission in the **Alerts and incidents** functional area is required to leave comments.

Assigning incidents to analysts

As a work item, an incident must be assigned to a SOC analyst or to a user group for inspection and possible investigation. You can assign incidents to a user group if you want to automatically assign incidents to analysts within the selected group. For details, refer to Configuring the automatic assignment of alerts and incidents.

You can change the assignee of an incident at any time.

Incidents can be assigned only to analysts and user groups who have the access right to read and modify alerts and incidents.

To assign one or several incidents:

1. In the main menu, go to **Monitoring** → **Incidents**.

2. Select the check boxes next to the incidents that you want to assign.

   You must select only the incidents detected in the same tenant. Otherwise, the **Assign to me** and **Assign** buttons will be disabled.

   Alternatively, you can assign an incident to an analyst or to a user group from the incident details. To open the incident details, click the link with the incident ID.

3. Do one of the following:

   - To assign the incidents to you, click the **Assign to me** button.

   - To assign the incidents to other analyst or user group:

     a. Click the **Assign** button.

     b. In the **Assign** window that opens, select whether to assign the incidents to a user or a user group.

     c. Start typing the name of an analyst or a user group, and then select the assignee from the list.

     d. Click the **Assign** button.

   If you want to unassign incidents, select **Not assigned**.

Incidents are assigned.

You also can assign an incident to an analyst by using playbooks.

Changing an incident status

As a work item, an incident has a status that shows the current state of the incident in its life cycle.

You can change the status of your own incidents or the incidents of other analysts only if you have the access right to read and modify alerts and incidents.

When you change the incident status, the incident and any unassigned linked alerts are automatically assigned to you.

If the incident status is changed manually, playbooks will not launch automatically. You can launch a playbook for such an incident manually.

An incident can have one of the following statuses:

- **New**

  When you create an incident or it is created automatically, the incident has the *New* status. You can change the status to *In progress* or *Closed*. When you change the *New* status to *Closed* and the incident has no assignee, the incident is automatically assigned to you.

- **In progress**

  This status means that an analyst started working on the incident or resumed the work by changing the *On hold* status. You can change the *In progress* status to any other status.

- **On hold**

  This status means that an analyst suspended work on the incident. Normally, you change the *On hold* status to *In progress* when the work is resumed, but you can change the *On hold* status to other statuses as well.

- **Closed**

  You close incidents when no additional work on the incident is expected. You can close an incident with one of the following resolutions:

  - True positive
  - False positive
  - Low priority

When you close an incident, the linked alerts also gain the *Closed* status and inherit the resolution from the incident. If the incident has no assignee, the closed incident is automatically assigned to you. If the closed incident has unassigned linked alerts, those alerts are automatically assigned to you.

The *Closed* status can only be changed to status *New*. If you want to return a closed incident back to work, change its status as follows: *Closed* → *New* → *In progress*.

To change status of one or several incidents:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. Do one of the following:
   - Select the check boxes next to the incidents whose status you want to change.
   - Click the link with the ID of the incident whose status you want to change.

     The **Incident details** window opens.
3. Click the **Change status** button.
4. In the **Change status** pane, select the status to set.

   When you select the *Closed* status, you must select a resolution.

   If at least one SLA metric was breached for an incident, the **SLA breach reasons** field appears when you close the incident. In this field, you need to specify the reason for the SLA breach.

   If you have selected the **Allow users with certain permissions only to close this incident** check box when editing the *Closed* status in the incident workflow, you must have either Main Administrator or Approver XDR role to close the incident.

   If you change the incident status to *Closed* and this incident contains uncompleted playbooks or response actions, all related playbooks and response actions will be terminated.
5. Click the **Save** button.

The status of the selected incidents is changed.

You also can change an incident status by using playbooks.

Changing an incident priority

As a work item, an incident has a priority that defines the order in which the incident must be investigated by analysts. You can change the incident priority manually.

You can change incident priorities of your own incidents or incidents of other analysts only if you have the access right to read and modify alerts and incidents.

An incident can have one of the following priorities:

- **Low**
- **Medium** (default value)
- **High**
- **Critical**

Incidents with the **Critical** priority are the most urgent ones and must be investigated first. The **Low** priority usually means that the incident is placed in the backlog. You can define your own criteria as to which priority should be set to which incident.

To change an incident priority:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. Do one of the following:
   - Select the check boxes next to the incidents whose priority you want to change.
   - Click the incident ID to open the details of the incident whose priority you want to change.
3. Click the **Change priority** button.
4. In the **Change priority** window, select the priority to set.
5. Click the **Save** button.

   The priority of the selected incidents is changed.

You also can change an incident priority by using playbooks.

Merging incidents

Two or more incidents may be interpreted as indicators of the same issue in an organization's IT infrastructure. If this is the case, you can merge the incidents to investigate them as a single issue.

When you merge incidents, you need to select a target incident among them. After the incident consolidation, the issue is to be investigated within the target incident. The target incident must have a status other than *Closed*. Other incidents are merged into the target one and, after consolidation, gain the *Closed* status and the **Merged** resolution.

All of the alerts linked to the merged incidents are automatically linked to the target incident. Because an incident can have no more than 1000 linked alerts, the application counts the alerts linked to the incidents that you want to merge. If the total number of linked alerts exceeds 1000, the selected incidents cannot be merged. When you select the target incident, a total number of alerts after merging is displayed.

> You cannot merge child incidents or incidents that have child incidents.

To merge incidents from the incident table:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. Select the check boxes next to the incidents that you want to merge into a target incident. You will select the target incident on the first step of the Wizard.
3. Click the **Merge into incident** button.
4. The **Merge incidents** Wizard opens.
5. Select the target incident.
6. Click the **Merge** button.

   The incidents are merged.

To merge incidents by using incident details:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. Click an incident ID to open the incident details. This incident will be merged into a target incident. You will select the target incident on the first step of the Wizard.
3. Click the **Merge into incident** button.

   The **Merge incidents** Wizard opens.
4. Select the target incident.
5. Click the **Merge** button.

   The incidents are merged.

Exporting incidents to a file

You can export incidents to a PDF or JSON file.

To export an incident to a PDF file:

1. In the main menu, go to **Monitoring** → **Incidents**. to invoke the incident table.
2. Click the ID of an incident you want to export.
3. Click the **Export to PDF** button in the window with incident details.

   A **Print** window opens.
4. In the **Print** window, expand the **Printer** list and select **Save as PDF**.
5. Click the **Save** button to save the incident as a PDF file.

   If you sort the incident table by **Type** or **Name** fields, the exported PDF file also uses this sorting.

To export all incidents to a JSON file:

1. In the main menu, go to **Monitoring** → **Incidents**. to invoke the incident table.
2. Click the **Export** button in the panel above the table.
3. Specify the file name and select the folder to which to export incidents.
4. Click **Save** to save all incidents as a JSON file.

Creating Managed Detection and Response incidents from XDR incidents

You can create a Managed Detection and Response incident from an XDR incident if Managed Detection and Response integration is enabled in OSMP and the incident's alerts meet the following conditions:

- The detection source is endpoint.
- The selected incident must have at least one alert that has a related asset.
- The detection technology is any of the following:
  - IOA rules
  - IOC
  - YARA
  - SB (Sandbox)
  - AM (Anti-Malware Engine)

Refer to the following article for information on how to enable Managed Detection and Response integration in OSMP: Integration with Kaspersky applications and solutions.

To export an incident to Managed Detection and Response:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident. The window with incident details is displayed.
3. Click the three-dots icon on the toolbar and select **Create MDR incident**.
4. In the **Create MDR incident** panel, specify the incident name and description and click **Create**.

   If the incident is created successfully, an appropriate message is displayed. You can now access the incident in the MDR Web Console. The link to the incident is available in the Summary section of the XDR incident.

Investigation graph

The *investigation graph* is a visual analysis tool that shows relationships between the following objects:

- Events
- Alerts
- Users
- Observables
- Assets (devices)
- Segmentation rules

The graph displays the details for an incident: the corresponding alerts and their common properties.

To open the investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident.

   The window with incident details is displayed.
3. Click the **View on graph** button.

> The **Write** permission in the **Alerts and incidents** functional area is required to view the graph. Refer to the following topic for details: Predefined user roles.

You can use the pan and zoom panel on the bottom right to navigate a complex graph.

Also, you can undo any action on the investigation graph by clicking the undo icon (↶), and revert a previously undone action by clicking the redo icon (↷). The maximum number of actions that you can undo or redo is 10.

If you close the investigation graph while performing actions in it, your history of changes is cleared. This means that after you open the investigation graph again, you cannot undo and redo the actions that you performed before closing the graph.

> When editing the investigation graph simultaneously with other users, you can view the changes made by the users, but you cannot undo and redo their actions.

Display mode

The investigation graph has two display modes: automatic, when the graph nodes are displayed along the timeline, and manual. By default, the automatic mode is applied.

The timeline is a scale in the lower part of the screen that graphically visualizes the objects constituting the incident's context, the time when these objects appeared, and the whole period during which they appeared.

The left end of the scale represents the start of the period: the earliest event from all alerts included in the incident or the earliest alert. The right end of the scale represents the end of the period: the latest event from all alerts included in the incident or the latest alert.

The duration of the whole period is calculated automatically after all alerts in the incident are analyzed. If a new object is added or deleted from the investigation graph, or when you generate a process tree, the timeline is automatically rearranged.

The interval that corresponds to one division on the scale () is dependent on the whole period duration. To view how the interval is calculated, **click this link**.

If the whole period is...	the division value is...
greater than or equal to 1 year	1 year
greater than or equal to 28 days but less than 1 year	1 month
greater than or equal to 1 week but less than 28 days	7 days
greater than or equal to 24 hours but less than 1 week	24 hours
greater than or equal to 1 hour but less than 24 hours	1 hour
greater than or equal to 10 minutes but less than 1 hour	5 minutes
from 100 seconds to 10 minutes	1 minute
from 1 second to 1.6 minutes	10 seconds
less than 1 second	100 milliseconds

In automatic mode, the place of nodes on the investigation graph is dependent on the date and time when the corresponding objects were detected and defined as follows:

- Assets are placed at the beginning of the scale.
- Alerts are placed in accordance with the time they were created.
- Events are placed in accordance with their timestamps.
- Observables are placed depending on the way they were added to the investigation graph:
  - When added via an alert node, the observable is placed in accordance with the alert creation time.
  - When added via an event node, the observable is placed in accordance with the event timestamp.

  The observables have the time and date as the alerts or events to which they are related.

When you hover a mouse cursor over a node, the date and time when the node was created is displayed.

The points where alerts or events are placed on the graph, correspond to the columnar elements (▮) on the scale. When you hover a mouse cursor over a columnar element, the information about the alerts and events, that were simultaneously detected in the incident, is displayed.

Automatic mode suggests the following limitations:

- You cannot move the graph nodes.
- The manually created links are not displayed.
- When one objects has several links (for example, one observable is linked to several alerts), only the link that comes from the oldest object (the one with the earliest creation time) is displayed on the graph.
- You cannot manually add objects which are not related to the automatically defined context of the incident.

  You can only add new objects through the context menu of the ones that are already displayed (events and observables related to an alert, or alerts and observables related to an event). After you switch the mode to manual, the object that you added in automatic mode is placed in close proximity to the node to which it is related. If the object is linked to several nodes, it must be placed in close proximity to the node with the latest timestamp.

If necessary, you can switch between the investigation graph modes by clicking the icon. In manual mode, the timeline will not be displayed, and you will be able to move the nodes.

> If in manual display mode you add an object that has no timestamp, the object will not be displayed after you the mode to automatic.

Interacting with graph nodes

You can use the toolbar at the top to add alerts and observables.

You can click and drag graph nodes to rearrange them.

You can click a graph node to bring the context menu.

Common context menu items:

- **View details**

  Opens a details window for the selected node.
- **Copy**

  Copies the node value to clipboard.
- **Hide**

  Removes the selected node from the graph.
- **Add comment/Edit comment**

  Invokes a text field for a comment. Click **Send** to confirm changes. Click **Cancel** to revert changes.

Event-specific context menu items:

Process tree

Only available for specific event types. Generates a process tree for the event. The blue color indication for an event indicates that you can generate a process tree for this event.

Alert-specific context menu items:

- **Change status**

  Invokes a **Change status** panel that allows you to change the alert status.

- **Events**

  A sub-menu that allows you to add events as graph nodes.

- **Observables**

  A sub-menu that allows you to add common observables as graph nodes.

- **Devices**

  A sub-menu that allows you to add common devices as graph nodes.

  By using this sub-menu, you also can view the list of alerts or incidents related to the device. To do that, in the sub-menu, click **View details** to open the **Device details** window, and then click the **Show related** button and select either **Alerts** or **Incidents**.

- **Users**

  A sub-menu that allows you to add users as graph nodes.

Observable-specific context menu items:

- **Find similar events**

  Invokes a **Threat hunting** panel that shows similar events.

- **Find similar alerts**

  Invokes an **Alerts** panel that shows similar alerts.

- **Request status from Kaspersky TIP**

  Allows you to obtain detailed information about the selected observable from Kaspersky Threat Intelligence Portal (Kaspersky TIP). Refer to the following topic for details: Integration with Kaspersky Threat Intelligence Portal.

- **Enrich data from Kaspersky TIP**

  Use this button to obtain detailed information about the selected observable from Kaspersky TIP. Refer to the following topic for details: Integration with Kaspersky Threat Intelligence Portal.

Segmentation rule-specific context menu items:

- **Edit in KUMA**

  Opens the KUMA Console in a new browser tab that displays the rule details.

- **Find similar alerts**

  Invokes an **Alerts** panel that shows similar alerts.

You can also add observables by clicking an alert or event. To do this, in the context menu that opens, you need to select **Observables**, and then click the observable. The observable will be added to the investigation graph. You can remove an observable from the investigation graph, if needed. To do this, you have to click the observable, and then click **Hide** in the context menu that opens.

Grouping graph elements

The investigation graph automatically groups alerts with common properties.

To ungroup an alert:

1. Click a graph element corresponding to an alert group.
   A table shows up that lists the alerts.
2. Select an alert that you want to show on the graph.
3. Click the **Show on graph** button in the table toolbar.
   The alert is added as a graph node.
4. Click the **Hide on graph** button, if you want to hide an alert.

Linking graph elements

The investigation graph automatically creates links for new items when applicable. Links can be added manually.

To manually add a link:

1. Click the **Link nodes** button.
   Link points appear around graph nodes.
2. Click and drag from a link point of one node to a link point of another node.

Manually created links have a color indication.

To remove a manually created link:

1. Hover over a manually created link.
   A cross icon appears in the middle of the link.
2. Click the cross icon to delete the link.

Threat hunting

You can analyze events to search threats and vulnerabilities that have not been detected automatically. To do this, you need to click the **Threat hunting** button in the toolbar at the top or invoke a graph node's context menu and click **Events** or **Find similar events**. The **Threat hunting** panel opens. Refer to the following section for details: Threat Hunting.

Exporting the graph

You can save the graph in the SVG format. To do this, click the **Export** button in the toolbar at the top.

Viewing and editing comments

You can leave comments for any graph node or link.

To add or edit a comment:

1. Click a graph element and select **Add comment** or **Edit comment**.
2. Click **Save** to save changes.

   The message ( ) icon appears next to the graph element. You can click the message ( ) icon to display the comment.

To display all comments,

Click the **Comments** button in the toolbar at the top. A panel with comments appears on the right. You can click a comment in the panel to highlight the corresponding message ( ) icon on the graph. You can use the panel to leave comments for the entire graph.

To delete a comment,

Select a comment on the graph or in the panel with comments. Click the ellipsis (⋮) icon and select **Delete**.

Process trees

During an incident investigation in Kaspersky Next XDR Expert, you can visualize the chain of events by using the process tree. The process tree displays parent and child processes to help you analyze the incident in more detail and track the exact sequence of events.

You can build and save up to five process trees for each event. A process tree is saved automatically once it is built. You can access saved process trees later from the **Threat hunting** section or from specific event's details.

Building process trees

You can build a process tree from the event or alert details, from the investigation graph, or from the Threat hunting section.

To build a process tree from event details:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident.

   The **Incident details** window is displayed.
3. Click the **View on graph** button.
4. Switch to **Manual** display mode.
5. Click the **Threat hunting** button.
6. In the **Threat hunting** panel, specify the parameters of the desired events.
7. Click the **Run query** button.

8. Click the row containing the desired event.
9. Click the **Build process tree** button.

   The process tree is built.

To build a process tree from the investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident.

   The **Incident details** window is displayed.
3. Click the **View on graph** button.
4. Switch to **Manual** display mode.
5. Right-click the event for which you want to build a process tree.
6. Click the **Show on graph** button.

   The process tree is built.

To build a process tree from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting**.
2. Specify your query and click the **Run query** button.
3. In the **Events** section, click the row containing the desired event.
4. Click the **View on graph** button.

   The process tree is built.

Deleting process trees

To delete a process tree:

1. Click the **View saved trees** button.
2. Click the button to the left of the process tree's name.
3. Click the **Delete** button.

   The process tree is deleted and cannot be restored.

Commenting on process trees

You can add comments to specific nodes or links in a process tree and provide a description of the tree. All comments are displayed in the **Comments** section.

To add a comment:

1. Right-click the node or link to which you want to add a comment.
2. Enter your comment.
3. Click the **Save** button.

To display all comments,

Click the **Comments** button in the toolbar at the top. A panel with comments appears on the right. You can click a comment in the panel to highlight the corresponding message ( ) icon on the process tree. You can use the panel to leave comments for the entire process tree.

To edit or delete a comment:

1. Click the comment.
2. Click the ellipsis (⋮) icon.
3. Select **Edit** or **Delete**.

Segmentation rules

Segmentation rules allow you to automatically split related alerts into different incidents based on the conditions that you specify when creating the rules.

Use segmentation rules to create different incidents based on related alerts. For example, you can combine several alerts with an important distinguishing feature into a separate incident.

Alerts can only be linked to an incident that belongs to the same tenant.

The segmentation rules are triggered according to the specified priority. The higher the rule is in the list of segmentation rules, the higher its priority. If you want to change the segmentation rule priority, drag and drop the rule by clicking its name.

We recommend that you use segmentation rules together with aggregation rules to define more precise rules for creating incidents.

To create a segmentation rule:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.
2. Click the tenant for which you want to create a segmentation rule.
3. In the **Settings** tab, click the **Segmentation rules** subsection of the **Detection and response** section.

   The list of segmentation rules opens.
4. Click the **Create** button.

   A **Create segmentation rule** window appears.

5. Specify the segmentation rule settings:

- **Status**

  Enable or disable the rule. By default, the rule is disabled.

- **Rule**

  A unique name for the rule.

- **Max alerts in incident**

  Maximum number of alerts in a single incident. The value must be between 1 and 200, and also must be greater than or equal to **Min alerts in incident**. The default value is 200. If the number of alerts exceeds the specified value, another incident is created.

- **Min alerts in incident**

  Minimum number of alerts in a single incident. The minimum and default value is 1. The value must be less then or equal to **Max alerts in incident**. If the number of alerts does not reach the specified value, an incident is not created.

- **Search interval**

  A time interval from which to select alerts and incidents. You can specify the value in days or in hours. The default value is 30 days.

- **Description**

  Optional. Rule description.

- **Groups**

  A jq expression that defines the array of string identifiers by which to assign alerts to incidents.

  If necessary, you can copy the jq expression for **Groups** section from another segmentation rule. To do that, click the **Copy from another rule** button.

  Example: `if any(.Observables[]? | select(.Type | ascii_downcase == "username") | .Value; startswith("adm_")) then ["Brute force admin"] else ["Brute force not admin"] end`

- **Incident name**

  A jq expression that defines the template for naming the incidents created according to this segmentation rule. By default, the following expression is specified: `"\(.Rules[]?.Name), \(.SourceCreatedAt)"`

  If necessary, you can copy the jq expression for **Incident name** section from another segmentation rule. To do that, click the **Copy from another rule** button.

  Example: `if any(.Observables[]? | select(.Type | ascii_downcase == "username") | .Value; startswith("adm_")) then "Brute force admin" else "Brute force not admin" end`

- **Trigger**

  A jq expression that defines the condition for including alerts in the incident.

  If necessary, you can copy the jq expression for **Trigger** section from another segmentation rule. To do that, click the **Copy from another rule** button.

  Example: `any(.Rules[]?; .ID == "123")`

6. Click the **Save** button.

The segmentation rule is saved and displayed in the table of segmentation rules. If necessary, you can edit the rule setting by clicking its name in the table.

You can use the `inSubnet` logical operator in jq expressions to test if the left operand (IP address) is in the subnet of the right operand (subnet).

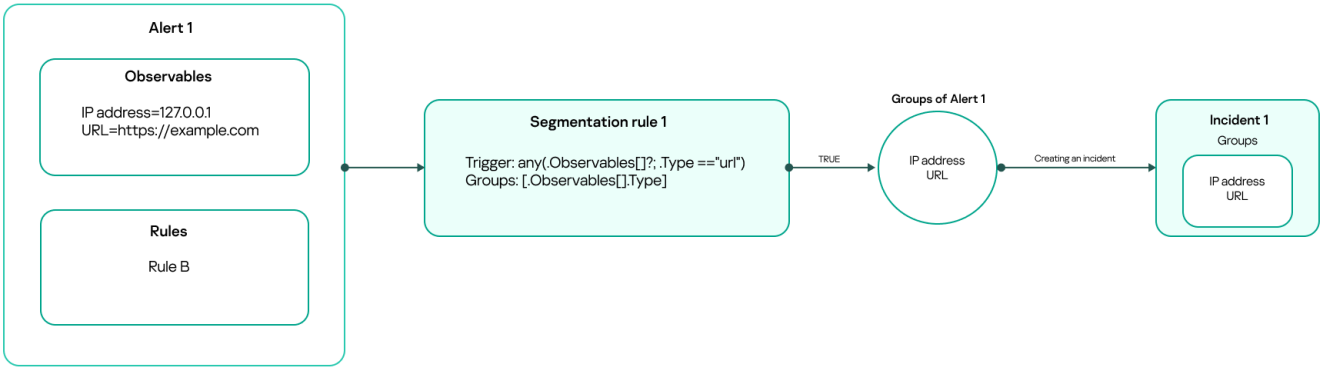
Example: `any(.Observables[]? | select(.Type == "ip") | .Value // empty; inSubnet("11.11.11.1/11"))`

The following predefined segmentation rules are available:

- Links to existing incident: Same MD5 file hash
- Links to existing incident: Same host asset
- Links to existing incident: Same IOA, IOC or Yara rule
- Create incident: Same MD5 file hash
- Create incident: Same host asset
- Create incident: Same IOA, IOC or Yara rule

The predefined rules are automatically applied to tenants.

When an alert is created, it is checked by all active segmentation rules in accordance with their priority. After the first rule is triggered, an array of string identifiers is formed for the alert, and the search starts for the incident to which the alert will be linked.

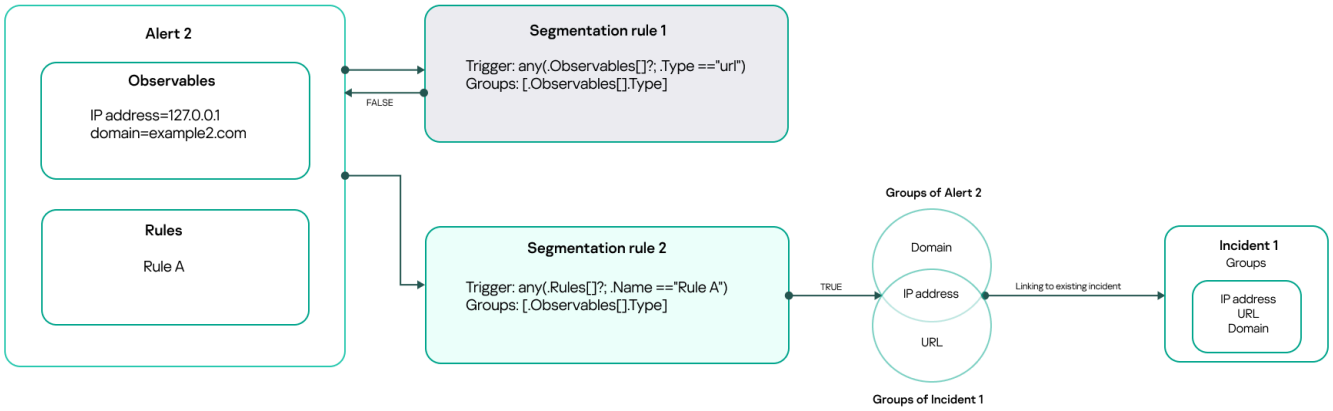


Creating a new incident after the first segmentation rule is triggered

A rule is triggered if the jq expression that you have specified in the **Trigger section** returns `true`.

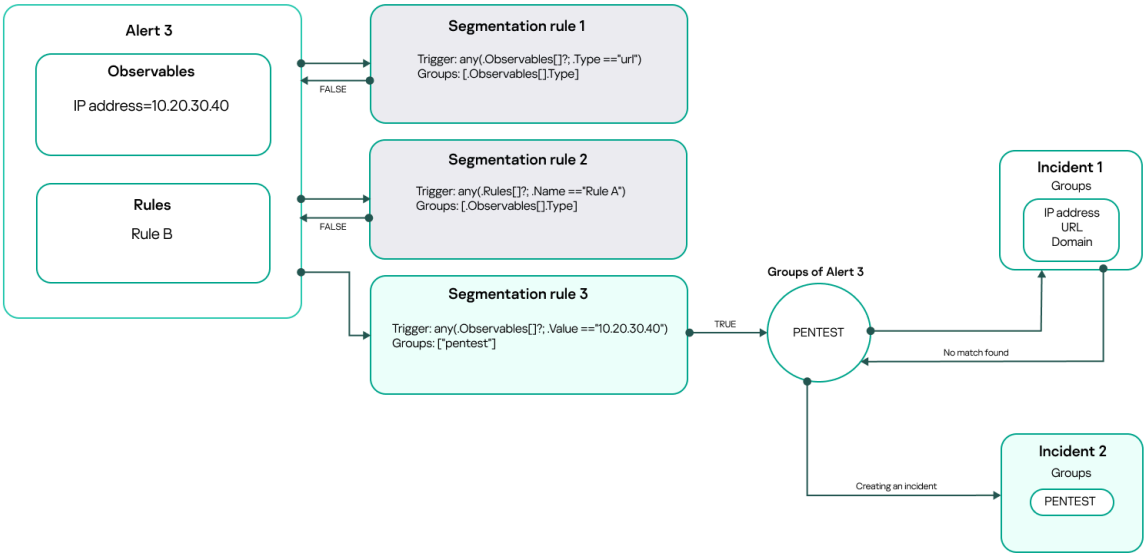
Alerts cannot be linked to incidents created manually.

An incident also has an array of string identifiers, which include the arrays of the alerts already linked to this incident. If the alert for which the segmentation rule was triggered has at least one element in its array that matches with any of those in the incident's array, the alert is linked to the incident. As a result, the array of this alert is added to the incident's array.



Linking an alert to an existing incident after detecting matching elements

If there are several incidents meeting the condition, the alert is linked to the one with the most recent update. If there are no incidents with matching elements in arrays, a new incident is created.



Creating a new incident after no matching elements are found based on the first two segmentation rules

When an incident is new, its array is empty. A new incident takes the array of string identifiers from an alert after the alert is linked.

Segmentation rule. Example

Configure the aggregation rules from the Aggregation rules. Example section in this topic.

The table below illustrates how to combine all penetration testing alerts in a single incident.

Segmentation rule

Attribute	Value
Trigger	.AggregationID == "Pentest"
Groups	["Pentest"]
Incident Name	"Pentest incident"

Aggregation and segmentation rules. Example

The table below illustrates how to combine alerts that have the same rule id in two incidents based on the user name prefix.

Aggregation rule

Attribute	Value	Description
Trigger	any(.Rules[]?; .ID == "123")	Searches alerts with the rule id set to "123".
Aggregation ID	if any(.OriginalEvents[]?.BaseEvents[]?.DestinationUserName // empty; startswith("adm_")) then "rule123_DestinationUserName_adm" else "rule123_DestinationUserName_not_adm" end	Searches for user names with the "adm_" prefix.
Alert Name	if any(.OriginalEvents[]?.BaseEvents[]?.DestinationUserName // empty; startswith("adm_")) then "Rule123 admin" else "Rule123 not admin" end	Sets the alert name depending on the user name prefix.

Segmentation rule

Attribute	Value
Trigger	.AggregationID | startswith("rule123_DestinationUserName")
Groups	[.AggregationID]
Incident Name	.Name

Copying segmentation rules to another tenant

You can copy an existing segmentation rule to another tenant.

> When a child tenant is created, it automatically copies all segmentation rules from the parent tenant. Editing segmentation rules in the parent tenant does not affect already created child tenants.

To copy segmentation rules:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the tenant that has the segmentation rule that you want to copy.
3. In the **Settings** tab, select **Segmentation rules**.
4. Select the segmentation rules you want to copy, and then click **Copy to tenant**.
5. Select one or multiple target tenants and click **Copy**.
6. If the target tenant contains a segmentation rule with an identical name, an **Overwrite or rename segmentation rules?** window appears.

Click **Overwrite** to delete the previously created rule for the target tenant and replace it with the rule that you want to copy.

Click **Copy and rename** to preserve the previously created rule and copy the specified rule with `(copy)` appended to its title.

> When you click the close (×) button, the window is closed, and the segmentation rules whose names are identical to the ones in the target tenants are not copied. However, if at step 4 you also selected the segmentation rules which are absent in the target tenants, such rules are copied.

The segmentation rules are copied to another tenant or tenants.

Managing Service Level Agreement (SLA)

Kaspersky Next XDR Expert allows you to configure Service Level Agreement (SLA) entries that define the time limits to process incidents.

You can define an SLA entry for a combination of incident workflow and priority.

The **Incidents** table and incident details show the remaining time for SLA entries.

You can configure an email notification template with the **Incident SLA expiring** and **Incident SLA changed** types.

Restarting calculation of SLA metrics when the incident status is reverted

When an incident is considered responded, SLA metrics for this incident stop being calculated. Reverting the incident status restarts the SLA metric by default.

To prevent restarting SLA metrics when the incident status is reverted:

1. Ensure that the kubectl utility is installed on the administrator host.
2. Run the following command:

   ```
   kubectl -n irp set env deploy/incident
   OSMP_INCIDENT_SLA_DISABLE_RESTART_CALCULATION_ON_START_TRIGGER=true
   ```

3. Restart the incident service.

Creating Service Level Agreement (SLA) entries

Creating SLA entryies requires one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator.

To create an SLA entry:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Click the **Settings** tab, and then click **Incident management**.

4. Select the **SLA** tab, and then click the **Create** button.

5. In the **Create SLA** pane that opens, configure the following settings:

   - **SLA name**—The display name of the entry.

   - **Priority**—The incident priority to which to apply the entry.

   - **Workflow**—The incident workflow to which to apply the entry.

     You can only assign a single entry to a combination of **Workflow** and **Priority**. You cannot select the incident workflow that are used in other SLAs with the same priority.

   - **Time to triage**—The time limit to accept the incident, in days, hours and minutes. The incident is considered accepted when its status changes from one status to another.

     To define this transition, select the **From status** and the **To status** values.

   - **Time to respond**—The time limit for initial response to the incident, in days, hours and minutes.

     Select stop condition:

     - If you want the incident to be considered responded after the response action is applied, select **On first response action**. After that, select one or multiple values in the **From status** field.

     - If you want the incident to be considered responded after its status changes to another one, select **When status changes to**. After that, select one or multiple values in the **From status** and the **To status** fields.

   - **Time to resolve**—The time limit to resolve the incident, in days, hours and minutes. The incident is considered resolved when its status changes from one status to another.

     To define this transition, select the **From status** and the **To status** values.

6. Click the **Create** button.

   The created SLA entry is displayed in the table.

Deleting Service Level Agreement (SLA) entries

Deleting SLA entries requires one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator.

To delete an SLA entry:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Click the **Settings** tab and then click **Incident management**.

4. Select the **Delete** tab.

5. Select the check box next to the SLA entry that you want to delete.

6. Click **Delete**.

7. In the confirmation window that opens, click **Delete**.

When you delete an SLA entry, time limits for incidents with the entry's **Incident type** and **Priority** are no longer calculated.

Managing incident types

Kaspersky Next XDR Expert allows you to manage incidents and customize the incident handling process by using incident types.

An incident type is a set of attributes, for which you can configure different processes, for example, assign a workflow to the incident type, configure a trigger, or configure a playbook algorithm.

You can create an incident type or use predefined incident types that you can customize.

Incident types can be active or inactive. If the incident type is active, you can select this type in the incident details window.

The incident type marked as a default type is assigned to all new incidents automatically. You cannot switch a default incident type to inactive.

The **Common** incident type is set as default. You can edit this setting.

You can create only one default incident type in a tenant.

Viewing the incident types table

To view the incident types table:

1. In the main menu, go to **Settings** → **Tenants**.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. On the **Settings** tab, click **Incident management**.

   The **Types** tab is displayed with the incident types table.

4. If you want to configure the incident types table, do any of the following:

   - Click the filter icon ( ), and then specify and apply the filter criterion in the invoked menu.
   - To hide or display a column, click the settings icon ( ), and then select the necessary column.

The incident types table contains the following information:

- **Name**. Name of the custom or predefined incident type.
  The table contains the following predefined incident types:

  - Common

    By default, this type has the **Yes** value in the **Default** column.

  - Information gathering
  - Compromise
  - Unauthorized access
  - Malware attack
  - Phishing
  - Availability
  - Insider threat
  - Data breaches
  - Configuration error
  - Supply chain attack
  - Web application attack
  - Vulnerability exploitation

- **Active type**. If the incident type is active, you will be able to select this type in the incident details window.
- **Default**. When you create an incident, the default type is automatically assigned to it. Possible values:

  - **True**
  - **False**

- **Workflow**. Incident workflow.
- **Tenant**. Name of the tenant to which the incident type belongs.

- **Creation type**. Way the incident type was created. Possible values:
  - **Custom**
  - **Predefined**
- **ID**. Unique identifier of the custom or predefined incident type. By default, this column is hidden.
- **Description**. Incident type description. By default, this column is hidden.

If necessary, you can create new incident types, as well as edit and delete predefined and custom incident types.

Creating incident types

To create an incident type:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.
   The tenant's properties window opens.
3. On the **Settings** tab, click **Incident management**, and then select the **Types** tab.
4. Click the **Create** button.
   The **Create incident type** window opens.
5. If you want the new incident type to be active, switch on the **Active type** toggle button.
6. In the **Name** field, enter the name of the new incident type.
7. If you want all new incidents to be assigned this type by default, select the **Set as default** check box.

   > There can be only one default incident type in a tenant. It means that if the tenant already has a default incident type, this type will no longer be default after you select this check box.

8. In the **Workflow** field, select the incident workflow.
9. If necessary, in the **Description** field, enter an incident type description or a comment.
10. Click the **Create** button.

    The new incident type is displayed in the incident types table.

Editing incident types

If necessary, you can edit incident types.

To edit an incident type:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.

   The tenant's properties window opens.
3. On the **Settings** tab, click **Incident management**.

   The **Types** tab is displayed with the incident types table.
4. Click the name of the incident type that you want to edit.

   The **Edit incident type** window opens.
5. Make your edits, and then click **Save**. For more details on the incident types properties that you can edit, refer to Creating incident types.

The incident type properties are edited and saved.

Deleting incident types

If you want to delete an incident type that is used in a playbook, you have to delete this incident type from the playbook trigger and/or algorithm to avoid errors.

You cannot delete an incident type in the following cases:

- An incident type is set as default in the tenant where this incident type was created.

  When trying to delete this incident type, you are prompted to set a new default incident type. In the window that opens, you have to select the incident type from the list.
- An incident type is set as default in a child tenant.
- The current tenant or a child tenant contains an incident with the type that you want to delete.

  Before deleting such a type, you have to assign another type to the incident.
- A Service Level Agreement (SLA) rule is defined for the incident type.

To delete the incident type:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.

   The tenant's properties window opens.
3. On **Settings**, click **Incident management**.

   The **Types** tab is displayed with the incident types table.

4. Do one of the following:

   - Select the incident type that you want to delete, and then click **Delete**.

   - Click the name of the incident type that you want to delete, and then in the **Edit incident type** window, click **Delete**.

5. In the confirmation dialog box, click **Delete**.

   The incident type is deleted.

Managing incident workflows

Kaspersky Next XDR Expert allows you to configure a flexible incident workflow. Kaspersky Next XDR Expert also visualizes the workflow in the visual editor.

The incident workflow is a set of statuses and transitions that an incident goes through during its lifecycle. *Status* is a step in the incident handling process. *Transition* helps the incident to move between different statuses. A transition is a link that allows you to configure transitions from one incident status to another and back. If necessary, you can use a transition as a one-way link.

You can create an incident workflow or use a predefined workflow that you can customize.

You also can assign a workflow to the incident types. This will help you manage the incident lifecycle in the most convenient way.

Viewing incident workflows table

To view the incident workflows table:

1. In the main menu, go to **Settings** → **Tenants**.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. On the **Settings** tab, click **Incident management**, and then select the **Workflows** tab.

   To hide or display a column, click the settings icon (⚙), and then select the necessary column.

   The incident workflows table is displayed.

The incident workflows table contains the following information:

- **Name**. Name of the custom or predefined incident workflow.

- **Linked types**. Number of linked incident types.

- **Tenant name**. Name of the tenant to which the incident workflow belongs.

- **Creation type**. Way the incident workflow was created. Possible values:
  - **Custom**.
  - **Predefined**.
- **Workflow ID**. Unique identifier of the incident workflow. By default, this column is hidden.
- **Description**. Incident workflow description.By default, this column is hidden.

Predefined incident workflows

Kaspersky Next XDR Expert allows you to manage incidents by using the predefined incident workflow. In the incident workflows table, such workflow is named **Standard**. In the **Creation type** column, these workflows are marked as **Predefined**.

If necessary, you can edit the predefined workflow to customize it.

The table below shows the statuses of the predefined workflow, and the reasons why incidents switch to these statuses.

Status	Reasons
New	• A new incident has been created (manually or automatically). • The incident status has been changed to **New** from one of the following statuses: **In progress**, **On hold**, or **Closed**.
In progress	The user manually changed the incident status from **New** or **On hold** to **In progress**.
On hold	The user manually changed the incident status from **In progress** to **On hold**.
Closed	• The user closed the incident. • The user linked the incident to another similar incident that has not been closed yet.

Creating incident workflows

The incident workflow allows you to manage incident lifecycle.

To create an incident workflow:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.

   The tenant's properties window opens.
3. On the **Settings** tab, click **Incident management**, and then select the **Workflows** tab.
4. Click the **Create** button.

   The **Create workflow** window opens.

   By default, each incident workflow contains predefined statuses **New** and **Closed**. You cannot delete or edit these statuses.

5. In the **Name** field, enter the name of the new workflow.

6. If necessary, in the **Description** field, enter a workflow description or a comment.

7. To add new statuses, in the **Workflow** section, click **Add status**.

8. In the window that opens, specify the following settings:

   a. In the **Status name** field, enter the name of the new status.

   b. In the **Category** field, select one of the following status categories:

      - **New**
      - **In progress**
      - **Resolved**
      - **Closed**

      The category determines the color of the status icon.

   c. In the **Incoming transition** field, select one or several incoming statuses.

      If you want to configure a transition from all statuses to the incoming statuses, select the **Allow all statuses to transition to this one** option.

   d. In the **Outgoing transition** field, select one or several outgoing statuses.

      If you want to configure a transition from the outgoing statuses to all statuses, select the **Allow this status to transition to all statuses** option.

   e. Click **Add**.

      The visualized workflow is displayed in the **Create workflow** window.

      If necessary, repeat steps 7-8e to add new statuses.

9. In the **Create workflow** window, click **Save**.

The new incident workflow is displayed in the table.

Editing incident workflows and statuses

You can edit workflow properties, as well as workflow' statuses and transitions.

To edit the incident workflow:

1. In the main menu, go to **Settings** → **Tenants**.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. On the **Settings** tab, click **Incident management**, and then select the **Workflows** tab.

4. Click the name of the workflow that you want to edit.

   The **Edit workflow** window opens.

5. Edit the workflow properties. For more details on the workflow properties that you can edit, see Creating incident workflows.

The workflow's properties are modified and saved.

To edit statuses of the incident workflow:

1. In the main menu, go to **Settings** → **Tenants**.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. On the **Settings** tab, click **Incident management**, and then select the **Workflows** tab.

4. Click the name of the workflow that you want to edit.

   The **Edit workflow** window opens.

5. Click the name of the status that you want to edit.

   The **Edit status** window opens.

6. Edit the status and transition settings. For more details on the status settings that you can edit, see Creating incident workflows.

   If necessary, you can delete the status by clicking the **Delete** button.

   > You cannot edit the name and the category of the following predefined statuses: **New** and **Closed** statuses. You also cannot delete these predefined statuses.

   > You cannot delete a status if it is assigned to an incident.

7. Click the **Save** button.

The workflow statuses are modified and saved.

Deleting incident workflows

> You cannot delete the incident workflow if there are linked incident types that belong to the parent or child tenant. In this case, you need to assign a different workflow to the linked incident types, and then try to delete incident workflow again.

> If you want to delete a workflow that is used in a playbook, before deleting, edit the playbook's trigger and/or algorithm to avoid errors.

To delete an incident workflow:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.

   The tenant's properties window opens.
3. On the **Settings** tab, click **Incident management**, and then select the **Workflows** tab.
4. In the list of workflows, select the workflow that you want to delete, and then click **Delete**.
5. In the confirmation dialog box, click **Delete**.

The incident workflow is deleted.

Managing incident effort

Kaspersky Next XDR Expert allows you to track effort for incident-related work. This helps calculate the time spent by an analyst on incident analysis and response, evaluate analysts' workload, and allocate labor resources efficiently.

Analysts can manually add details of their effort on incident-related work, including time spent and, if necessary, effort category.

Effort categories can be viewed, added, edited or deleted through the API. To manage effort categories, you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator.

To manage incident effort, you must have one of the following XDR roles: Main administrator, Tenant administrator, Junior analyst, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver.

Kaspersky Next XDR Expert also allows you to generate and export an effort report by analysts or groups for further analysis.

Adding effort for incidents

You can add the effort spent on incident-related work to the **Effort log** section of the incident details. This helps analyze the time analysts spend on handling incidents, plan their workload, and calculate payroll costs.

To add effort for an incident:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the incident table, click the ID of the required incident.

   The window with incident details is displayed.
3. Go to the **Effort log** tab, and then click the **Add** button.

   The **Add effort** pane opens.

4. If necessary, in the **Category** drop-down list, select the effort category.

5. In **Hours** and **Minutes** fields, specify the time you spent on incident-related work.

   Maximum value for the **Minutes** field is 59.

6. Click the **Add** button.

   The effort you spent on incident-related work is added.

Editing effort for incidents

You can edit previously added efforts.

You cannot edit effort added by other analysts.

To edit effort for an incident:

1. In the main menu, go to **Monitoring** → **Incidents**.

2. In the incident table, click the ID of the required incident.
   The window with incident details is displayed.

3. Go to the **Effort log** tab.

4. In the **Date** column, click the link with the date and time for the effort that you want to edit.
   The **Edit effort** pane opens.

5. Edit the effort properties, and then click the **Save** button.

   The effort properties are modified and saved.

Deleting effort for incidents

You can delete previously added efforts.

You cannot delete effort added by other analysts.

To delete effort for an incident:

1. In the main menu, go to **Monitoring** → **Incidents**.

2. In the incident table, click the ID of the required incident.
   The window with incident details is displayed.

3. Go to the **Effort log** tab.

4. In the list of efforts, select the check box next to the effort that you want to delete, and then click the **Delete** button.

The effort is deleted.

Generating and exporting incident reports

You can generate and export the reports on analysts' effort spent to work with incidents and on the incident compliance to SLA metrics. The reports are saved in the CSV format.

To generate a report for an incident, you must have:

- Access to the tenant to which the incident belongs.
- One of the following XDR roles: Main administrator, SOC administrator, Tenant administrator, Junior analyst, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Auditor.

To generate and export a report:

1. In the main menu, go to **Monitoring** → **Incidents**.
2. In the toolbar, click the **Export** button, and then, in the drop-down menu, select one of the following:
   - **Effort report** to export a report on the effort spent by analysts when working with an incident.
   - **Incident metrics report** to export a report on the incident compliance to SLA metrics.
3. In the right pane that opens, specify the period for which you want to display the data in the report. To do this, select the date and time in the **From** and **To** fields.

   By default, in the **From** field the first day of the current month is selected for **Effort reports**, and the current date is selected for **Incident metrics reports**. The time is set to 00:00 for both types of reports.

   In the **To** field, the current date is selected, and the time is set to 23:59.
4. If necessary, specify the other settings, depending on the type of report that you selected at step 3:
   - For the **Effort report**:
     - Select analysts or user groups if you want the report to include the incidents that required the efforts of the users that you specify.
     - Enable the **Include incident ID and effort category details** toggle switch if you want the effort report to contain these details.

       By default, the toggle switch is disabled.

- For the **Incident metrics report**:
  - Select the priorities of the incidents to be included in the report. Possible values: **Low**, **Medium**, **High**, **Critical**.
  - Enable the **Export only incidents with SLA breaches** toggle switch if you want to include in the report only the incidents with SLA breaches.

    An incident is included in the report if it has at least one expired SLA entry. The SLA entry is considered expired when at least one of its lime limits (**Time to triage**, **Time spent to respond**, or **Time to resolve**) is less than 0.

5. Click the **Export** button.

   The pane closes, and the report is generated and downloaded to your computer.

Configuring the retention period of alerts and incidents

Kaspersky Next XDR Expert allows you to reduce or increase the retention periods of alerts and incidents, depending on your needs. By default, the retention period of alerts and incidents is 360 days.

> The child tenant copies the retention period of alerts and incidents from the parent tenant. If necessary, you can edit the retention period for the child tenant.

To configure the alert or incident retention period:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.

   The tenant's properties window opens.
3. On the **Settings** tab, click **Alerts & incidents**.
4. In the **Retention period** section, specify the new retention period in one or both of the following fields:
   - **Alert retention period (days)**
   - **Incident retention period (days)**

   The minimum value is `1`.
5. Click **Save**.

   The new retention period is configured.

Regardless of the configured retention period, if the expired alert is linked to an unexpired incident, the alert will be deleted only after the retention period of the linked incident expires. If the expired incident has unexpired linked alerts, the incident will be deleted only after the retention period of the linked alerts expires.

If the expired parent incident has the unexpired child incidents, the parent incident will be deleted only after the retention period of the last child incident expires.

Configuring the automatic assignment of alerts and incidents

Kaspersky Next XDR Expert allows you to configure the automatic assignment of alerts, incidents, and child incidents to an analyst. If automatic assignment is enabled, alerts or incidents that are already assigned to a user group manually or by using a playbook will be automatically assigned to an analyst from that group. The analyst who is assigned an alert or incident must meet the following conditions:

- Have the **Available** work status.
- Have the fewest active alerts and incidents in the group.
- Have a user session duration equal to or longer than the configured minimum.

If multiple analysts meet the criteria for automatic assignment, the application assigns the alert or incident to one of them at random.

If there are no users in the group assigned to the alert or incident, or if no eligible users (for example, all users have the **Busy** status), the alert or incident will remain assigned to the group.

The automatic assignment runs only once, at the moment the alert or incident is assigned to the user group. If the application cannot assign the alert or incident to a user during this initial attempt, the alert or incident remains assigned to the group. No further attempts to assign the alert or incident are made.

To configure automatic assignment, you must have one of the following XDR roles: Main administrator, SOC administrator, Tenant administrator.

To configure the automatic assignment of alerts and incidents:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the name of the required tenant.

   The tenant's properties window opens.
3. On the **Settings** tab, click **Alerts & incidents**.
4. In the **Automatic assignment** section, enable the automatic assignment for one or more options below:
   - **Alerts**
   - **Incidents**
   - **Child incidents**

5. In the **Min duration of user activity (minutes)** field, specify the minimum duration of an analyst's user session.

   Once this value is reached, the analyst is included in the list of users eligible for alert and/or incident assignment. By default, the value is `5`.

6. Click **Apply**.

The automatic assignment is configured.

After an alert or incident is automatically assigned to a user, the assignment information will be displayed in the alert or incident details under the **History** section. When an alert or incident is automatically assigned, the log records **System** as the author.

Viewing asset details

Asset details are a window that contains all information related to the asset.

You can view asset details in one of the following ways:

- From alert details
- From incident details
- From event details (if the event contains assets)
- From an investigation graph

To view asset details:

1. In the main menu, go to **Monitoring**, and then do one of the following:

   - If you want to view asset details from alert details, click **Alerts**, and then in the **ID** column, click the ID of the alert that includes the asset whose details you want to view. In the window that opens, go to the **Assets** tab.
   - If you want to view asset details from incident details, click **Incidents**, and then in the **ID** column, click the ID of the incident that includes the asset whose details you want to view. In the window that opens, go to the **Assets** tab.
   - If you want to view asset details from event details, click **Threat hunting**, and then click the event that contains the asset whose details you want to view.
   - If you want to view asset details from an investigation graph, click **Incidents**, and then in the **ID** column, click the ID of the incident that includes the asset whose details you want to view. In the window that opens, click the **View on graph** button. Click the alert node, and then in the context menu, select **Devices** or **Users**.

2. Do one of the following:

   - Click the name of the required asset, and then in the drop-down list, select **View properties**.
   - On the investigation graph, click the asset node, and then in the context menu, select **View details**.

The asset details window is displayed.

In the asset details window, you can switch between the **General** and **Prevention rules** tabs.

General tab

The **General** tab contains the following sections:

- **Asset properties**—Information about the asset, for example, asset name, ID, and tenant to which the asset belongs.

  If the current license in KUMA includes the AI module, the **AI score** field is displayed and shows the asset score. This field shows the degree of atypical activity on the asset and may take the values in the range from 0 to1, where 0 is expected behavior, and 1 is completely unexpected behavior for the asset within the infrastructure. If Kaspersky Endpoint Security with a built-in agent is installed on the asset (device), the **Isolation status** field is displayed. You can view and change the network isolation status of the device by clicking the **Settings** link next to the status.

- **Categories**—Information about the categories associated with the asset.

- **Custom fields**—Asset fields that you created in KUMA Console.

- **Installed software**—Information about software installed on the asset.

- **KSC: vulnerabilities**—Vulnerabilities of the asset, if provided. This information is available for assets imported from Kaspersky Security Center.

- **Kaspersky applications**—Information about the Kaspersky applications installed on the asset.

- **Device protection status**—Status of the asset; the following values are possible: *OK*, *Critical*, *Warning*. This information is available for assets imported from Kaspersky Security Center.

- **KICS for Networks: asset properties**—Information about the asset. This information is imported from KICS for Networks.

- **KICS for Networks: vulnerabilities**—Vulnerabilities of the asset, if provided. This information is available for assets imported from KICS for Networks.

You can expand the sections by clicking the chevron icons (>) next to their names.

From the toolbar, you can perform the following actions:

- Edit the asset details by clicking the **Edit** button.

- Run a response action by clicking the **Response** button, and then selecting a response action from the drop-down list.

- Assign a category to the asset by clicking the **Link to category** button.

- Move to the **Response history** section, the **Threat hunting** section, the list of related alerts, or the list of related incidents by clicking the **View in** button, and then from the drop-down list, selecting where you want to move.

  The **Response history** section, the events in the **Threat hunting** section, the alerts or incident will be filtered by the asset from whose details you moved.

- Connect to the asset by using the remote terminal. To do this, click the **Open remote terminal** button.
- Export information about the asset to a CSV file by clicking the **Export to CSV** button.

Prevention rules tab

The tab contains the table of prevention rules applied to the device.

The table is displayed with one of the following statuses of the rules progress: **Applied, Applying**, or **Error**.

> To view the prevention rules list you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator, SOC manager, Tier 1 analyst, Tier 2 analyst, Junior analyst, Approver, Auditor.

The table has the same columns as the one that you open from the **Resources and services** section.

You can view and edit the prevention rule settings. To do this, open the prevention rule details by clicking the link with the rule name in the **Name** column. The **Edit prevention rule** pane opens with the **Details** tab selected. You can edit the prevention rule settings.

Also, you can click the **Go to Threat hunting** button to view the events obtained when a prevention rule was triggered. On the **Change log** tab, you can view all the changes applied to the prevention rule.

> To open the prevention rule details, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.
> If the prevention rule belongs to a parent tenant to which you have no rights, you cannot edit the prevention rule.

Investigating and responding via remote terminal

Kaspersky Next XDR Expert allows you to connect to an asset by using the remote terminal. Once connected, you can interact with the operating system shell in real time to investigate alerts and incidents, perform diagnostics, and respond to threats.

Through the remote terminal, you can manage the asset by using the commands supported by the operating system installed on the asset.

By using the remote terminal, you can connect to multiple assets simultaneously. In this case, a separate remote terminal window is opened for each asset in OSMP Console.

You can connect to the asset by using the remote terminal if the following conditions are met:

- You have activated the application under the Kaspersky Next XDR Expert license.
- You have one of the following XDR roles: Main administrator, Tier 1 analyst, Tier 2 analyst.
- Kaspersky Endpoint Security for Linux or Kaspersky Endpoint Security for Windows is installed on the asset.

You cannot use the remote terminal to connect to the asset from which you access the OSMP Console.

If an asset is compromised, managing this asset through the remote terminal can lead to unintended execution of malicious code. Use the remote terminal only to connect to devices you trust.

Managing remote terminal connections

You can use the remote terminal to investigate threats and respond to them on the assets in real time.

You can access the remote terminal in one of the following ways:

- From the incident details.
- From the alert details.
- From the asset details.
- From the asset table (**Detection and response** tab).

Connecting to assets by using the remote terminal

To connect to assets:

1. In the main menu, do one of the following:
   - To access the remote terminal from the incident details, go to **Monitoring** → **Incidents**. In the incident table, click the ID of the required incident, and then, in the window that opens, go to the **Assets** tab.
   - To access the remote terminal from the alert details, go to **Monitoring** → **Alerts**. In the alert table, click the ID of the required alert, and then, in the window that opens, go to the **Assets** tab.
   - To access the remote terminal from the asset details, go to **Asset management** → **Assets (Devices)**. Select the **Detection and response** tab, and then, in the asset table, click the name of the required asset. In the context menu, click **Details**. In the asset details pane that opens, click the ellipsis (⋮) icon.
   - To access the remote terminal from the asset table, go to **Asset management** → **Assets (Devices)**. Select the **Detection and response** tab, and then, in the asset table, select the required asset.
2. Click the **Open remote terminal** button.

3. If you want to perform response actions on the asset, perform the following steps:

   a. In the **Remote terminal** window that opens, run one of the following:

      - For devices running Windows:

        ```
        response.exe
        ```

      - For devices running Linux:

        ```
        response
        ```

   b. Press **`Enter`**.

   c. Copy the password by clicking the **Copy password** link, and then paste the password into the remote terminal.

After the remote session with the asset has been established, you can investigate threats and perform response actions if needed.

All actions performed in the remote terminal are logged.

If there is no activity for more than 10 minutes, the connection to the asset is terminated.

Viewing active connections in the remote terminal

With the remote terminal, you can connect to and work with multiple assets at the same time. If you have not manually ended a session with an asset, or it has not expired due to inactivity, the session remains active even if you navigate to another section of OSMP Console.

To view to the active remote terminal sessions,

in the upper-right corner of OSMP Console, click the **N connections**.

The **Remote terminal** window opens.

Disconnecting from assets by using the remote terminal

To disconnect from assets,

in the **Remote terminal** window, click the cross (✕) icon to disconnect from the asset.

Alternatively, if you want to disconnect from more than one asset, click the **Disconnect all** button.

The asset is disconnected from the remote terminal.

Performing response actions via remote terminal

After you connected to the asset via the remote terminal, you can investigate threats and respond to them by using command line interface.

To run commands, use the following syntax:

```
<common> <parameter> <argument> [--option=value] [--flag]
```

Note that `--flag` is boolean flag. If specified, it is set to `true`. You can also set it explicitly to `true` or `false`. If not specified, it is set to `false`. The `--case-sensitive` flag is an exception to this rule, and its behavior is described in more detail below on this page.

Common command line parameters

Parameter	Type	Description
help	string	Open online help.
task-timeout	integer	Task execution time limit (seconds).
storage-path	string	Path to the file share for storing response action results.

Response action parameters

You can use the following parameters depending on the response action you want to perform on the asset:

Delete file

To delete a file from the device, use the following response action parameters:

Parameter	Arguments	Options	Type	Description
`delete-file`	[filePath]	`md5`	string	MD5 file hash. Used with the file path. Can be specified with or without SHA256.
		`sha256`	string	SHA256 file hash. Used with the file path. Can be specified with or without MD5. Available only on devices running Windows.
		`skip`	array of enum	Conditions for skipping the response action. Possible values: • `self-defence` • `sco` • `kl-signed` • `kl-by-trusted-app` Available only on devices running Windows.
		`with-subfolders`	flag	Perform a recursive search if `filePath` specifies a directory.
		`case-sensitive`	flag	Indicates whether the path is case-sensitive. Available only on devices running Windows. If the `--case-sensitive` flag is not specified, the task uses the case-sensitivity behavior of the operating system and runs as case-sensitive. To run the task as case-insensitive, specify `--case-sensitive=false`.

Example

```
delete-file "C:/file path.txt" --md5=12345 --case-sensitive --skip=self-defence --skip=sco --skip=kl-signed --skip=kl-by-trusted-app
```

Manage service

To manage a service on the device, use the following response action parameters:

Parameter	Arguments	Options	Type	Description
`change-service`	[operationType] Possible values: • `start` • `stop` • `pause` • `resume` • `delete` • `change-startup-type`	`startup-type`	enum	Specifies the startup type for the `change-startup-type` operation. This parameter is ignored for other operations. Possible values: • `Boot` • `System` • `Automatic` • `Manual` • `Disabled` Available only on devices running Windows.
		`service-name`	string	Service name. Masks are supported. Available only on devices running Windows.
		`md5`	string	MD5 file hash. Can be specified with or without SHA256. Available only on devices running Windows.
		`sha256`	string	SHA256 file hash. Can be specified with or without MD5. Available only on devices running Windows.

Example

```
change-service change-startup-type --startup-type=auto --service-name=ServiceName --md5=12345
```

Move file to quarantine

To prevent a threat distribution, you can move an infected file to quarantine. To do that, use the following response action parameters:

Parameter	Arguments	Options	Type	Description
`quarantine-file`	[filePath]	`originator`	enum	Response action initiator. Possible values: • `edr-user` • `edr-ioc-scan` • `ksb-ioc-scan` • `mdr-backup` • `kata-edr` • `kics-edr` Available only on devices running Windows.
		`skip`	array of enum	Conditions for skipping the response action. Possible values: • `self-defence` • `sco` • `kl-signed` • `kl-by-trusted-app` Available only on devices running Windows.
		`legacy-file-id`	flag	When specified, returns legacy 64-bit quarantine IDs instead of GUIDs.
		`case-sensitive`	flag	Specifies whether the path is case-sensitive. Available only on devices running Windows. If the `--case-sensitive` flag is not specified, the task uses the case-sensitivity behavior of the operating system and runs as case-sensitive. To run the task as case-insensitive, specify `--case-sensitive=false`.
		`delete`	flag	Deletes the file after quarantining it.
		`md5`	string	MD5 file hash. Used with the file path. Can be specified with or without SHA256.
		`sha256`	string	SHA256 file hash. Used with the file path. Can be specified with or without MD5.

Example

```
quarantine-file C:/filePath --originator=edr-user --skip=self-defence --skip=sco --skip=kl-signed --skip=kl-by-trusted-app
```

Delete file from quarantine

To delete an infected file from the quarantine directory on the device, use the following response action parameters:

Parameter	Options	Type	Description
delete-quarantined-file	file-id	string	ID of the quarantined file.
	legacy-file-id	integer	Legacy quarantine ID.

Example

```
delete-quarantined-file --file-id=453C4D2A-1B45-49C9-A954-1C086E4347F0
```

Restore file from quarantine

If you have reasons to believe that the file in the quarantine is safe, you can restore the file on the device. To do that, use the following response action parameters:

Parameter	Options	Type	Description
restore-quarantined-file	file-id	string	ID of the quarantined file.
	legacy-file-id	integer	Legacy quarantine ID
	replace	flag	Replaces the existing file.
	remove	flag	Removes the file from quarantine after restoration.
	destination	string	Specifies the destination path used to restore the object from backup. If not specified, the object is restored to the original path.
	fallback-destination	string	Specifies a fallback destination path. Used only if restoring to the original path fails because the path does not exist or access is denied.

Example

```
restore-quarantined-file --file-id=453C4D2A-1B45-49C9-A954-1C086E4347F0 --replace --fallback-
destination="C:/path"
```

Get file from quarantine

The quarantined file located on a device can be obtained from the quarantine directory for further analysis.To get the file from the quarantine, use the following response action parameters:

Parameter	Options	Type	Description
get-quarantined-file	file-id	string	ID of the quarantined file.
	legacy-file-id	integer	Legacy quarantine ID.
	stream-index	integer	Index of the file stream to retrieve. Available only on devices running Windows.

After the response action is completed, the obtained file is displayed in the **Files** section of the OSMP Console.

Example

```
get-quarantined-file --file-id=453C4D2A-1B45-49C9-A954-1C086E4347F0 --stream-index=0
```

Terminate process

To terminate processes on the device, use the following response action parameters:

Parameter	Options	Type	Description
`terminate-process`	`path`	string	Path to the process executable.
	`pid`	integer	System process ID.
	`unique-pid`	integer	Process ID. Available only on devices running Windows.
	`md5`	string	MD5 file hash. Used with the file path. Can be specified with or without SHA256.
	`sha256`	string	SHA256 file hash. Used with the file path. Can be specified with or without MD5.
	`case-sensitive`	flag	Specifies whether the path is case-sensitive. If the `--case-sensitive` flag is not specified, the task uses the case-sensitivity behavior of the operating system and runs as case-sensitive. To run the task as case-insensitive, specify `--case-sensitive=false`.
	`expand-env`	flag	Expands environment variables in paths. Available only on devices running Windows.
	`skip`	array of enum	Conditions for skipping the response action. Possible values: • `self-defence` • `sco` • `kl-signed` • `kl-by-trusted-app` Available only on devices running Windows.
	`limit`	integer	Maximum number of process termination attempts. If not specified, the number of attempts is unlimited. Available only on devices running Windows.

Example

```
terminate-process --pid=1234 --limit=100
```

Get file

To get a file from the device, use the following response action parameters:

Parameter	Arguments	Options	Type	Description
`get-file`	[filePath]	`md5`	string	MD5 file hash. Used with the file path. Can be specified with or without SHA256.
		`sha256`	string	SHA256 file hash. Used with the file path. Can be specified with or without MD5.
		`max-size`	integer	Maximum file size in bytes. The response action fails if the file exceeds this size. By default, the size is unlimited.
		`case-sensitive`	flag	Specifies whether the path is case-sensitive. Available only on devices running Windows. If the `--case-sensitive` flag is not specified, the task uses the case-sensitivity behavior of the operating system and runs as case-sensitive. To run the task as case-insensitive, specify `--case-sensitive=false`.

After the response action is completed, the file is saved to a password-protected ZIP archive. The password is *infected*.

Example

```
get-file C:/file --md5=1234567890 --max-size=1000 --case-sensitive
```

Enable network isolation

To enable network isolation for the device, use the following response action parameters:

Parameter	Options	Type	Description
`start-network-isolation`	`timeout`	integer	Duration of network isolation. When the timeout expires, isolation is disabled automatically. By default, the duration is unlimited.
	`expand-env`	flag	Expands environment variables in paths. Available only on devices running Windows.
	`no-notify-user`	flag	Disables user notification about the network isolation operation. Available only on devices running Windows.
	`exclude`		Network exclusion rule. This option cannot be used independently. To configure exclusions, specify one or more child options: • `process` • `direction` • `remote-ip-v4` • `remote-ip-v6` • `local-port-from` • `local-port-to` • `remote-port-from` • `remote-port-to` • `protocol`
	`process`	string	Specifies an optional receiving or transmitting process. The value must be the full path to the executable. Masks are not supported. Environment variables are supported on Windows only. Available only on devices running Windows.
	`direction`	enum	Specifies the connection direction for connection-based protocols (for example, TCP) or the packet direction for datagram-based protocols (for example, UDP). If not specified, both directions are used. Possible values: • `inbound` • `outbound` • `both`
	`remote-ip-v4`	string	Remote IPv4 address in the xxx.xxx.xxx.xxx format. Subnet prefixes are supported. If neither IPv4 nor IPv6 is specified, any IP address is matched.
	`remote-ip-v6`	string	Remote IPv6 address in the xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx format. Subnet prefixes are supported. If neither IPv4 nor IPv6 is specified, any IP address is matched. Available only on devices running Windows.
	`local-port-from`	integer	Lower bound of the local port range. If not specified, any port is matched. If the port is not applicable to the packet protocol, the rule is inactive.
	`local-port-to`	integer	Upper bound of the local port range. If not specified, any port is matched. If the port is not applicable to the packet protocol, the rule is inactive.
	`remote-port-from`	integer	Lower bound of the remote port range. If not specified, any port is matched. If the port is not applicable to the packet protocol, the rule is inactive.
	`remote-port-to`	integer	Upper bound of the remote port range. If not specified, any port is matched. If the port is not applicable to the packet protocol, the rule is inactive.
	`protocol`	integer	IANA protocol identifier (for example, TCP = 6). If not specified, any protocol is matched.

Example

```
start-network-isolation --timeout=300 exclude --process="C:/process.exe" exclude --local-port-from=1 --local-port-to=10 --direction=inbound --no-notifyuser
```

Disable network isolation

To disable network isolation for a device, use the following response action parameters:

Parameter	Options	Type	Description
`stop-network-isolation`	`no-notify-user`	flag	Disables network isolation for a device. By default, the user receives a notification when network isolation is disabled. If the `--no-notify-user` option is specified, the user is not notified. The `--no-notify-user` option is available only on devices running Windows.

Example

```
stop-network-isolation --no-notify-user
```

Get forensics

Getting a file list

To get a list of files and folders stored on the disk, use the following response action parameters.

Parameter	Options	Type	Description
`get-forensic files`	`case-sensitive`	flag	Specifies whether the path is case-sensitive. Available only on devices running Windows. If the `--case-sensitive` flag is not specified, the task uses the case-sensitivity behavior of the operating system and runs as case-sensitive. To run the task as case-insensitive, specify `--case-sensitive=false`.
	`with-data-streams`	flag	Specifies whether Alternate Data Streams are included in the forensic report. Available only on devices running Windows.
	`max-depth`	integer	Maximum search depth. 0 means unlimited depth. 1 searches only the specified directory. Available only on devices running Windows.
	`max-files`	integer	Maximum number of files or Alternate Data Streams to find. The search stops when the limit is reached. Available only on devices running Windows.
	`include-dir`	array of string	Directories to include in the scan. Directory paths can contain environment variables (for example, %TMP% or %TEMP%). If not specified, all local directories are included. Available only on devices running Windows.
	`exclude-dir`	array of string	Directories to exclude from the scan. Directory paths can contain environment variables (for example, %TMP% or %TEMP%). Available only on devices running Windows.
	`include-files`	array of string	File name masks to include in the search (for example, *.ps1). If not specified, all files are included. Available only on devices running Windows.
	`exclude-files`	array of string	File name masks to exclude from the search (for example, *.ps1). If not specified and `exclude-dir` is specified, all files are excluded. Available only on devices running Windows.

Example

```
get-forensic files --max-depth=3 --max-files=1000 --include-dir="C:/include1" --include-dir=D:/include2
```

Getting a processes list

To get a list of processes running on the device, run the following command:

```
get-forensic processes
```

Getting a list of autorun points

To get a list of autorun points, run the following command:

```
get-forensic autoruns
```

Getting a process memory dump

To get a process memory dump, use the following response action parameters:

Parameter	Options	Type	Description
`get-forensic process-dump`	`pid`	integer	System process ID.
	`unique-pid`	integer	Unique process ID. Available only on devices running Windows.
	`md5`	string	MD5 file hash. Used with the file path. Available only on devices running Windows.
	`sha256`	string	SHA256 file hash. Used with the file path. Available only on devices running Windows.

Example

```
get-forensic process-dump --pid=123 --md5=abcdef123456
```

Getting a file dump

To get a file dump, use the following response action parameters:

Parameter	Options	Type	Description
`get-forensic files-dump`	`case-sensitive`	flag	Specifies whether the path is case-sensitive. By default, this is determined by the file system. Available only on devices running Windows. If the `--case-sensitive` flag is not specified, the task uses the case-sensitivity behavior of the operating system and runs as case-sensitive. To run the task as case-insensitive, specify `--case-sensitive=false`.
	`path`	array of string	Full paths of files to dump. Generic and metadata files (for example, $Mft, $Boot) are supported. Masks are not supported.
	`with-data-streams`	flag	Specifies whether Alternate Data Streams are included in the forensic report. Available only on devices running Windows.
	`sha256`	string	SHA256 file hash. Used with the file path. Available only on devices running Windows.

Example

```
get-forensic files-dump --path="C:/path1"--path="D:/path2" --with-data-streams
```

Getting a registry key

To get a registry key, use the following response action parameters:

Parameter	Options	Type	Description
`get-forensic registry-dump`	`path`	array of string	Full paths of registry keys to dump. Masks are not supported.

Example

```
get-forensic registry-dump --path="C:/path1" --path="D:/path2"
```

Getting system information

To get system information of the device, run the following command:

```
get-forensic system-info
```

Available only on devices running Windows.

Getting environment variables

To get environment variables, run the following command:

```
get-forensic environment-variables
```

Getting a memory dump

To get a memory dump, use the following response action parameters:

Parameter	Options	Type	Description
`get-forensic memory-dump`	`compress`	flag	Specifies whether the dump is compressed using gzip with a fast compression level. Available only on devices running Windows.
	`segment-size`	integer	Specifies the size of dump segments in bytes. The dump is split into segments of the specified size. If the value is 0 or not specified, the dump is not split. Available only on devices running Windows.

Example

```
get-forensic memory-dump --compress --segment-size=1000
```

Getting a disk image

To get a disk image, use the following response action parameters:

Parameter	Options	Type	Description
`get-forensic disk-dump`	`logical`	string	Logical drive name in the OS-specific format (for example, C:, D:, E: on Windows). Available only on devices running Windows.
	`physical`	string	Physical drive name in the OS-specific format (for example, \\.\PHYSICALDRIVE0 on Windows). Available only on devices running Windows.
	`compress`	flag	Specifies whether the dump is compressed using gzip with a fast compression level. Available only on devices running Windows.
	`segment-size`	integer	Specifies the size of dump segments in bytes. The dump is split into segments of the specified size. If the value is 0 or not specified, the dump is not split. Available only on devices running Windows.

Example

```
get-forensic disk-dump --logical="C:" --segment-size=1000 --compress
```

Threat hunting

The **Threat hunting** page contains tools that help you analyze events to search threats and vulnerabilities that have not been detected automatically. To create an alert from a set of events, select the events, and then click the **Create alert** button.

You can open the **Threat hunting** page in any of the following ways:

- In the main menu, go to **Monitoring** → **Threat hunting**.
- In the Alert or Incident details, invoke the context menu for an attribute, and then select **Search in Threat Hunting**.
- In the Incident details, click the **View on graph** button. In the investigation graph that opens, click the **Threat hunting** button.

The **Threat hunting** page displays events. You can filter out events:

- By editing the SQL query
- By changing the time range
- By selecting the tenants to which the events belong

Working with events

The **Threat hunting** section contains tools that help you search threats and vulnerabilities by analyzing the events.

Event data model

Endpoint Agents send information about the following events to OSMP:

- Process
- File Create
- File Modify
- File Rename
- File Read
- File Delete
- File Attributes Modify
- File Creation Time Modify

- File Hardlink Created
- File Symlink Created
- Module
- Registry Create
- Registry Delete
- Registry Modify
- Registry Rename
- Registry Query
- Registry Save
- Connection
- Port Listen
- AMSI Scan
- Process Console Input
- Driver
- Process Interpreted File Run
- DNS
- Pipe Create
- Pipe Connect
- LDAP
- WMI Activity
- WMI Consumer Register
- Code Injection
- Process Access
- Threat Detect
- Threat Detect Processing Result
- Linux Event Log
- Windows Event Log
- Blocked Document

- Applock
- Process Terminated
- Device
- File System Mount
- File System Unmount
- Bits Job Create
- Bits Job Complete
- Bits Job Start
- Bits Job Stop
- Bits Job Add File
- Bits Job Error
- Service Create
- Service Modify
- Service Delete
- Scheduled Task Create
- Scheduled Task Modify
- Scheduled Task Delete
- File Owner Change
- File ACL Change
- Registry Owner Change
- Registry ACL Change
- Executable Memory Mapped
- Agent Start
- Agent Stop
- Agent Install
- Agent Uninstall
- Agent Action
- System Startup

- System Shutdown
- Connection Close

> Enabling the `Connection Close` event may result in a significant increase in resource usage on protected devices.

Process

This event type is supported on Windows, Linux, and macOS operating systems.

Process event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version

Field	Type	Description
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessCalltrace	string	Stack trace of the process
ProcessFileLnkName	string	Name of the LNK file that was used for the process execution (without the path)
ProcessFileLnkPath	string	Path to the LNK file that was used for the process execution (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was used for the process execution
ProcessFileLnkCmdline	string	Command line of the LNK file for the process execution
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file

Field	Type	Description
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS

Field	Type	Description
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id

Field	Type	Description
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file

Field	Type	Description
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Create

This event type is supported on Windows, Linux, and macOS operating systems.

File Create event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert

Field	Type	Description
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS

Field	Type	Description
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS

Field	Type	Description
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the file that was executed
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id

Field	Type	Description
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process.
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Creation flags of grand-parent process
GrandParentProcessDirectory	string	Current working directory of grand-parent process. Specified for Linux and macOS
GrandParentProcessFileLnkName	string	Name without the path of the LNK file that started the grand-parent process
GrandParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the grand-parent process
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process.
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified

Field	Type	Description
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working directory of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS

Field	Type	Description
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process.
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process.
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process.
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process

Field	Type	Description
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Modify

This event type is supported on Windows, Linux, and macOS operating systems.

File Modify event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version

Field	Type	Description
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags

Field	Type	Description
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file

Field	Type	Description
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)

Field	Type	Description
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process

Field	Type	Description
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process

Field	Type	Description
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file

Field	Type	Description
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Rename

This event type is supported on Windows, Linux, and macOS operating systems.

File Rename event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic

Field	Type	Description
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file

Field	Type	Description
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
FileOldName	string	Previous name of the file
FileOldPath	string	Previous path of the file
FileOldFullname	string	Previous full path to the file
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified

Field	Type	Description
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process

Field	Type	Description
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment

Field	Type	Description
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field

Field	Type	Description
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Read

This event type is supported on Windows and Linux operating systems (on Linux only for versions with eBPF integration).

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content

Field	Type	Description
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process

Field	Type	Description
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process

Field	Type	Description
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file

Field	Type	Description
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field

Field	Type	Description
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process

Field	Type	Description
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Delete

This event type is supported on Windows, Linux, and macOS operating systems.

File Delete event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC

Field	Type	Description
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed

Field	Type	Description
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID

Field	Type	Description
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)

Field	Type	Description
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file

Field	Type	Description
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Attributes Modify

This event type is supported on Windows and Linux operating systems.

File Attributes Modify event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file

Field	Type	Description
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS

Field	Type	Description
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process

Field	Type	Description
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified

Field	Type	Description
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS

Field	Type	Description
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process

Field	Type	Description
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Creation Time Modify

This event type is supported on Windows operating systems.

File Creation Time Modify event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version

Field	Type	Description
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileDelafterreboot	string	True if file will be deleted after reboot. Specified for delete operation. Specified only for Windows
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)

Field	Type	Description
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process

Field	Type	Description
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process

Field	Type	Description
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file

Field	Type	Description
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Hardlink Created

This event type is supported on Windows operating systems.

File Hardlink Created event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server

Field	Type	Description
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileTargetName	string	Name of the target file
FileTargetPath	string	Path of the target file
FileTargetFullname	string	Full path to the target file
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory

Field	Type	Description
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file

Field	Type	Description
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed

Field	Type	Description
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file

Field	Type	Description
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Symlink Created

This event type is supported on Windows operating systems.

File Symlink Created event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id

Field	Type	Description
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileTargetName	string	Name of the target file
FileTargetPath	string	Path of the target file
FileTargetFullname	string	Full path to the target file
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process

Field	Type	Description
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file

Field	Type	Description
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Module

This event type is supported on Windows and Linux operating systems.

Module event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id

Field	Type	Description
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileAttr	int	The file system attributes
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileSignSubjectname	string	Information about the organization that signed the file
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileType	string	Type of the file
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
FileHijackingpath	string	Path to original DLL
DotnetFileAssemblyName	string	Fully qualified assembly name. Contains certificate hash (used for filtering)
DotnetFileAssemblyFlags	int	.NET assembly flags

Field	Type	Description
DotnetFileModuleFlags	int	.NET assembly module flags
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username

Field	Type	Description
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file

Field	Type	Description
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry Create

This event type is supported on Windows operating systems.

Registry Create event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped

Field	Type	Description
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegValueName	string	Name of registry value
RegValueType	string	Type of registry value
RegValueData	string	Data in the registry value
RegValueFileMd5 [Reserved for future use]	string	MD5 hash of the file from the RegValueData field
RegValueFileFullname [Reserved for future use]	string	Full path to the file from the RegValueData field
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed

Field	Type	Description
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process

Field	Type	Description
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)

Field	Type	Description
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified

Field	Type	Description
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry Delete

This event type is supported on Windows operating systems.

Registry Delete event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegValueName	string	Name of registry value
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file

Field	Type	Description
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process

Field	Type	Description
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file

Field	Type	Description
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry Modify

This event type is supported on Windows operating systems.

Registry Modify event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack

Field	Type	Description
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegValueName	string	Name of registry value
RegValueType	string	Type of registry value
RegValueData	string	Data in the registry value
RegValueFileMd5 [Reserved for future use]	string	MD5 hash of the file from the RegValueData field
RegValueFileFullname [Reserved for future use]	string	Full path to the file from the RegValueData field
RegOldValueData	string	Previous data in the registry value
RegOldValueMd5	string	The previous MD5 hash value of the file from the RegValueData field
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file

Field	Type	Description
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process

Field	Type	Description
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process

Field	Type	Description
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileCapEffective	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username who executed process that runs child process.

Registry Rename

This event type is supported on Windows operating systems.

Registry Rename event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version

Field	Type	Description
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	
HostOsmpId	string	
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegValueName	string	Name of registry value
RegValueType	string	Type of registry value
RegValueData	string	Data in the registry value
RegOldKeyName	string	Previous registry key name
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process

Field	Type	Description
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process

Field	Type	Description
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file

Field	Type	Description
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry Query

This event type is supported on Windows operating systems.

Registry Query event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegValueName	string	Name of registry value
RegValueType	string	Type of registry value
RegValueData	string	Data in the registry value
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed

Field	Type	Description
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)

Field	Type	Description
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file

Field	Type	Description
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry Save

This event type is supported on Windows operating systems.

Registry Save event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack

Field	Type	Description
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegValueName	string	Name of registry value
RegValueType	string	Type of registry value
RegValueData	string	Data in the registry value
FileFullname	string	Full path to the file
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process

Field	Type	Description
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)

Field	Type	Description
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded

Field	Type	Description
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Connection

This event type is supported on Windows, Linux, and macOS operating systems.

Connection event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic

Field	Type	Description
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
NetSrcIp	string	Source ipv4/ipv6 address
NetSrcPort	int	Source port number
NetDstIp	string	Destination ipv4/ipv6 address
NetDstPort	int	Destination port number
NetUrlHostname	string	Remote host name
NetUrlPort	int	Remote host port
NetUri	string	Destination URI for HTTP
NetHttpMethod	string	HTTP method
NetHttpUseragent	string	An identifier of the client program that initiated the request
NetHttpReferrer	string	URI reference to the resource from which the target URI was obtained
NetProtocol	string	IP protocol
NetDirection	string	Direction of network connection (inbound, outbound)
NetHttpHeaderName	string	Name of HTTP header
NetHttpHeaderValue	string	Value of HTTP header
NetRaw	string	Flag whether the raw socket was used
NetTlsVersion	string	TLS version
NetTlsSni	string	TLS Server Name Indication
NetTlsSniEncrypted	string	TLS Encrypted server name indication
NetTlsCertificateMd5	string	MD5 hash of TLS certificate
NetTlsCertificateSha1	string	SHA1 hash of TLS certificate
NetTlsCertificateSubjectname	string	TLS Subject alternative name (SAN)
NetTlsCertificateIssuername	string	Issuer name of TLS certificate

Field	Type	Description
NetTlsCertificateSerialnumber	string	Serial number of TLS certificate
NetTlsCertificateCheckresult	string	Result of TLS certificate verification
NetTlsCertificateValidfrom	int	TLS certificate start-of-validity date
NetTlsCertificateValidto	int	TLS certificate end-of-validity date
NetTlsCiphersuite	int	TLS supported algorithms
NetTlsJa3	string	Ja3 fingerprint of TLS certificate
NetTlsJa3s	string	Ja3s fingerprint of TLS certificate
NetTlsJa3Md5	string	MD5 hash of Ja3 fingerprint of TLS certificate
NetTlsJa3sMd5	string	MD5 hash of Ja3s fingerprint of TLS certificate
ProcessDynamicstate	int	Terminated - 0x1, Remote threads created - 0x2, Remote memory writes possible - 0x4, Has visible windows - 0x8
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified

Field	Type	Description
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process

Field	Type	Description
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process

Field	Type	Description
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field

Field	Type	Description
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Port Listen

This event type is supported on Windows and Linux operating systems.

Port Listen event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
NetDstIp	string	Source ipv4/ipv6 address
NetDstPort	int	Source port number
NetProtocol	int	IP protocol
ProcessDynamicstate	int	Terminated - 0x1, Remote threads created - 0x2, Remote memory writes possible - 0x4, Has visible windows - 0x8

Field	Type	Description
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name

Field	Type	Description
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file

Field	Type	Description
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

AMSI Scan

This event type is supported on Windows operating systems.

AMSI Scan event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped

Field	Type	Description
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
AmsiContentType	string	Type of object that was sent to AMSI scan.
AmsiContentData	string	Content of the detected object.
AmsiContentName	string	Name of the object that was sent to AMSI scan.
AmsiContentMd5	string	MD5 of object that was sent to AMSI scan.
AmsiContentSha256	string	SHA256 of object that was sent to AMSI scan.
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file

Field	Type	Description
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process

Field	Type	Description
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file

Field	Type	Description
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Process Console Input

This event type is supported on Windows and Linux operating systems.

Process Console Input event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped

Field	Type	Description
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
InputText	string	Interactive input text
InputType	string	Type of input object
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level

Field	Type	Description
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process

Field	Type	Description
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment

Field	Type	Description
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field

Field	Type	Description
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process

Field	Type	Description
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Driver

This event type is supported on Windows operating systems.

Driver event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows,)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event

Field	Type	Description
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment

Process Interpreted File Run

This event type is supported on Windows, Linux, and macOS operating systems.

Process Interpreted File Run event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)

Field	Type	Description
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	The name of the file without path, that will be executed by the interpreter
FilePath	string	Path to the file without its name, that will be executed by the interpreter
FileFullname	string	Full path to the file, that will be executed by the interpreter
FileSize	int	Size (in bytes) of the file, that will be executed by the interpreter
FileCreationtime	int	Creation time of the file, that will be executed by the interpreter
FileModificationtime	int	Time of last modification of the file, that will be executed by the interpreter
FileAttrModificationtime	int	Time of the last change of the file attribute, that will be executed by the interpreter
FileMd5	string	MD5 hash of the file, that will be executed by the interpreter
FileSha256	string	SHA256 hash of the file, that will be executed by the interpreter
FileAttr	int	The list of file system attributes for file, that will be executed by the interpreter
FileType	string	Type of the file, that will be executed by the interpreter
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
FileProductname	string	The name of the application from the VERSIONINFO resource
FileProductvendor	string	Company name value from the VERSIONINFO resource
FileDescription	string	Description from the VERSIONINFO resource

Field	Type	Description
FileOriginalname	string	Original file name from the VERSIONINFO resource
FileSignSubjectname	string	The subject name of the certificate which was used to sing the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded, that will be executed by the interpreter
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment, that will be executed by the interpreter
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes

Field	Type	Description
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed

Field	Type	Description
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded

Field	Type	Description
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field

Field	Type	Description
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

DNS

This event type is supported on Windows and Linux operating systems (on Linux only for external connections).

DNS event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components

Field	Type	Description
DnsServerIp	string	DNS server being queried
DnsQueryOptions	string	Options of the performed DNS query
DnsQueryStatus	int	Error code for an operation
DnsResourceName	string	URL being queried
DnsResourceType	string	Type of performed DNS query
DnsAnswer	string	DNS query result
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded

Field	Type	Description
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS

Field	Type	Description
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process

Field	Type	Description
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file

Field	Type	Description
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Pipe Create

This event type is supported on Windows operating systems.

Pipe Create event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert

Field	Type	Description
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
PipeName	string	Name of the named pipe which created or connection have been established with
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file

Field	Type	Description
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id

Field	Type	Description
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id

Field	Type	Description
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process

Field	Type	Description
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process

Field	Type	Description
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Pipe Connect

This event type is supported on Windows operating systems.

Pipe Connect event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC

Field	Type	Description
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
PipeName	string	Name of the named pipe which created or connection have been established with
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment

Field	Type	Description
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file

Field	Type	Description
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified

Field	Type	Description
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

LDAP

This event type is supported on Windows operating systems.

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
LdapSearchScope	string	The area or range of the directory tree that should be searched
LdapSearchFilter	string	Conditions or criteria that an LDAP search query will use to match entries in a directory
LdapSearchDn	string	Starting point or base in the directory tree where the search operation will begin
LdapSearchAttr	string	List of specific attributes that are used for filtering or searching by
ProcessUniqueid	int	Process unique id

Field	Type	Description
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id

Field	Type	Description
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)

Field	Type	Description
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process

Field	Type	Description
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

WMI Activity

This event type is supported on Windows operating systems except for Windows XP.

WMI Activity event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack

Field	Type	Description
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
WmiRemote	string	Flag of remote WMI operation
WmiHostName	string	Name of the source host of WMI activity
WmiUserName	string	Name of the user account that performed the WMI operation
WmiNamespacename	string	WMI namespace that was queried
WmiQuery	string	WMI query
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags

Field	Type	Description
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process

Field	Type	Description
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)

Field	Type	Description
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

WMI Consumer Register

This event type is supported on Windows operating systems except for Windows XP.

WMI Consumer Register event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
WmiNamespacename	string	WMI namespace that was queried
WmiFiltername	string	Name of the filter used by the event consumer
WmiConsumerName	string	Name of the created event consumer
WmiConsumerText	string	MOF code of the event consumer
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed

Field	Type	Description
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process

Field	Type	Description
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file

Field	Type	Description
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Code Injection

This event type is supported on Windows and Linux operating systems.

Code Injection event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped

Field	Type	Description
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
TargetProcessUniqueid	int	Target process unique id
TargetProcessId	int	Target process identifier (PID)
TargetProcessCmdline	string	Command line arguments passed into the target process
TargetProcessFileName	string	Name of the executable file of the target process (the process in which the code injection was performed)
TargetProcessFilePath	string	Full path without the file name to the executable file of the target process (the process in which the code injection was performed)
TargetProcessFileFullname	string	Full path to the executable file of the target process (the process in which the code injection was performed)
TargetProcessFileMd5	string	MD5 of the executable file of the target process
TargetProcessFileSha256	string	SHA256 of the executable file of the target process
TargetProcessIntegritylevel	int	Target process integrity level
TargetProcessFileUrl	string	URL from where the target process executable file was downloaded
TargetProcessFileEmailSrcAddress	string	The sender's address from which the target process executable file was received as an attachment
SyscallArgs	string	System call arguments. Specified for Linux
ProcessCalltrace	string	Stack trace of the process
ProcessModifiedcmdline	string	Command line of the process

Field	Type	Description
InjectMethod	string	Injection method
InjectAddress	string	Field contains information about address from which execution starts after injection
InjectImageFileName	string	Only for SET_WINDOWS_HOOK
InjectImageFileFullname	string	Only for SET_WINDOWS_HOOK
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment

Field	Type	Description
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS

Field	Type	Description
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field

Field	Type	Description
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified

Field	Type	Description
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Process Access

This event type is supported on Windows operating systems.

Process Access event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
AccessOperationtype	string	The type of access operation performed on a process handle (by one process on another process).
AccessRights	string	Desired access rights to the process object
ProcessCalltrace	string	Stack trace of the process
TargetProcessFileName	string	Name of the executable file of the target process (the process in which the code injection was performed)
TargetProcessFilePath	string	Full path without the file name to the executable file of the target process (the process in which the code injection was performed)
TargetProcessFileFullname	string	Full path to the executable file of the target process (the process in which the code injection was performed)
TargetProcessFileMd5	string	MD5 of the executable file of the target process
TargetProcessFileSha256	string	SHA256 of the executable file of the target process
TargetProcessFileCreationtime	int	Creation time of the target process executable file
TargetProcessFileModificationtime	int	The last modification time of the target process executable file
TargetProcessFileAttrModificationtime	int	The last time, when the target process executable file attributes were modified

Field	Type	Description
TargetProcessFileSize	int	Target process executable file size in bytes
TargetProcessUniqueid	int	Target process unique id
TargetProcessId	int	Target process identifier (PID)
TargetProcessCmdline	string	Command line arguments passed into the target process
TargetProcessFileUrl	string	URL from where the target process executable file was downloaded
TargetProcessFileEmailSrcAddress	string	The sender's address from which the target process executable file was received as an attachment
HandleProcessSrcFileName	string	Name of the process from which the handle was duplicated
HandleProcessSrcFilePath	string	Full path without the file name of the process from which the handle was duplicated
HandleProcessSrcFileFullname	string	Full path of the process from which the handle was duplicated
HandleProcessSrcFileMd5	string	MD5 of the process from which the handle was duplicated
HandleProcessSrcFileSha256	string	SHA256 of the process from which the handle was duplicated
HandleProcessSrcFileCreationtime	int	Creation time of the process from which the handle was duplicated
HandleProcessSrcFileModificationtime	int	The last modification time of the process from which the handle was duplicated
HandleProcessSrcFileAttrModificationtime	int	The last modification time of attributes of the process from which the handle was duplicated
HandleProcessSrcFileSize	int	Size of the executable file of the process from which the handle was duplicated
HandleProcessSrcUniqueid	int	Unique PID of the process from which the handle was duplicated
HandleProcessSrcId	int	Process identifier (PID) of the process from which the handle was duplicated
HandleProcessSrcIntegritylevel	int	Integrity level of the process from which the handle was duplicated
HandleProcessSrcCmdline	string	Command line options of the process from which the handle was duplicated
HandleProcessSrcFileUrl	string	URL from where the source process was downloaded
HandleProcessSrcFileEmailSrcAddress	string	The sender's address from which the source process executable file was received as an attachment
HandleProcessDstFileName	string	Name of the process to which the handle was duplicated
HandleProcessDstFilePath	string	Full path without the file name of the process to which the handle was duplicated
HandleProcessDstFileFullname	string	Full path of the process to which the handle was duplicated
HandleProcessDstFileMd5	string	MD5 of the process to which the handle was duplicated
HandleProcessDstFileSha256	string	SHA256 of the process to which the handle was duplicated
HandleProcessDstFileCreationtime	int	Creation time of the process to which the handle was duplicated
HandleProcessDstFileModificationtime	int	The last modification time of the process to which the handle was duplicated
HandleProcessDstFileAttrModificationtime	int	The last modification time of attributes of the process to which the handle was duplicated
HandleProcessDstFileSize	int	Size of the executable file of the process to which the handle was duplicated
HandleProcessDstUniqueid	int	Unique PID of the process to which the handle was duplicated
HandleProcessDstId	int	Process identifier (PID) of the process to which the handle was duplicated
HandleProcessDstIntegritylevel	int	Integrity level of the process to which the handle was duplicated
HandleProcessDstCmdline	string	Command line options of the process to which the handle was duplicated
HandleProcessDstFileUrl	string	URL from where the source process was downloaded
HandleProcessDstFileEmailSrcAddress	string	The sender's address from which the source process executable file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id

Field	Type	Description
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource

Field	Type	Description
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file

Field	Type	Description
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Threat Detect

This event type is supported on Windows, Linux, and macOS operating systems.

Threat Detect event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic

Field	Type	Description
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
ThreatObjectFullname	string	Full path to the file
ThreatObjectMd5	string	MD5 hash of the file content
ThreatObjectSha256	string	SHA256 hash of the file content
ThreatObjectType	string	Type of the detected object
ThreatVerdictname	string	Threat verdict name
ThreatRecordid	int	The ID of the record that detected a threat
ThreatDatabaseTimestamp	int	Date and time when the database was last updated
ThreatProcessingmode	int	The processing mode at which the object was detected
ThreatSilent	string	True for silent detects
NetProtocol	int	IP protocol
NetDstIp	string	Local (attacked) ipv4/ipv6 address
NetDstPort	int	Local (attacked) port number
NetSrcIp	string	Remote (attacker's) ipv4/ipv6 address
NetSrcPort	int	Remote (attacker's) port number
ThreatEmailAddress	string	The sender's address from which the detected file was received as an attachment
ThreatEmailDestination	string	The destination address of the detected email
ThreatEmailSubject	string	The subject of the detected email
ThreatEmailMessageid	string	The Message-ID of the detected email
ThreatEmailReceivedtimestamp	string	The received time of the detected email
ThreatDownloaderFileFullname	string	The path to the executable file of the process that downloaded the detected file
AmsiContentType	string	Type of object that was sent to AMSI scan.
AmsiContentData	string	Content of the detected object.

Field	Type	Description
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name

Field	Type	Description
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file

Field	Type	Description
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Threat Detect Processing Result

This event type is supported on Windows, Linux, and macOS operating systems.

Threat Detect Processing Result event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped

Field	Type	Description
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
ThreatObjectFullname	string	Full path to the file
ThreatObjectMd5	string	MD5 hash of the file content
ThreatObjectSha256	string	SHA256 hash of the file content
ThreatObjectType	string	Type of the detected object
ThreatVerdictname	string	Threat verdict name
ThreatRecordid	int	The ID of the record that detected a threat
ThreatDatabaseTimestamp	int	Date and time when the database was last updated
ThreatProcessingmode	int	The processing mode at which the object was detected
NetProtocol	int	IP protocol
NetDstIp	string	Local (attacked) ipv4/ipv6 address
NetDstPort	int	Local (attacked) port number
NetSrcIp	string	Remote (attacker's) ipv4/ipv6 address
NetSrcPort	int	Remote (attacker's) port number
ThreatEmailAddress	string	The sender's address from which the detected file was received as an attachment
ThreatDownloaderFileFullname	string	The path to the executable file of the process that downloaded the detected file
ThreatAction	string	Type of action taken with the object that is detected by exact detect

Field	Type	Description
ThreatActionresult	string	The result of the taken action, specified in the field ThreatAction.
ThreatActionreason	string	In case of successful action, field contains 0. In case of unsuccessful action, this field contains the reason of failure
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID

Field	Type	Description
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)

Field	Type	Description
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file

Field	Type	Description
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Linux Event Log

This event type is supported on Linux operating systems.

Linux Event Log event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
LinuxEventType	string	Type of the event
LinuxEventMessage	string	Message from a log
LinuxOperationResult	string	Operation result—For example, Success or Failed
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed

Field	Type	Description
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource

Field	Type	Description
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file

Field	Type	Description
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS

Field	Type	Description
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags

Field	Type	Description
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file

Field	Type	Description
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Windows Event Log

This event type is supported on Windows operating systems.

Windows Event Log event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id

Field	Type	Description
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
SystemLogEventXml	string	XML from System Event Log
SystemLogEventId [inclusive filtering field, the name is constant]	int	Event ID in the system log
SystemLogChannelName [inclusive filtering field, the name is constant]	string	Channel name in the system log
SystemLogProviderName [inclusive filtering field, the name is constant]	string	Provider name in the system log
OmitQualifiers [inclusive filtering field, the name is constant]		
WinBinary	string	File attributes of an object being processed
WinData	string	Main data of the Windows Event Log event
WinAccessGranted	string	Security parameters from windows event log events
WinAccessList	string	Access rights for the object being processed
WinAccessMask	string	Access rights for the object being processed
WinAccessReason	string	Reason for moving the file to the category
WinAccessRemoved	string	Access rights that were removed from the object
WinAccountDomain	string	Name of the domain used to authenticate the owner of the logon session
WinAccountExpires	string	Active directory objects parameters
WinAccountName	string	Id of the account under which the controlled process was started
WinAccountSessionIdentifier	string	Authentication session identifier in the Network Policy Server event
WinAction	string	System or security action in the event
WinActionID	string	Identifier of the action performed during the event
WinActionName	string	Name of the action performed during the event
WinActiveProfile	string	Active configuration or network profile
WinAddedCAPs	string	List of added Central Access Policies
WinAdditionalActionsID	string	Identifiers of additional actions related to the event

Field	Type	Description
WinAdditionalActionsString	string	Description or strings of additional actions related to the event
WinAdditionalDetails	string	Query string
WinAdditionalInfo	string	Additional information about the event or object
WinAdditionalInfo2	string	Additional information about the event or object (second field)
WinAdvancedOptions	string	Advanced configuration options
WinAlgorithmName	string	Name of the cryptographic algorithm used
WinAllowedToDelegateTo	string	Id of the account under which the controlled process was started
WinAppCorrelationID	string	Object type code
WinApplication	string	Name of the object being processed
WinAppName	string	Software name
WinAppVersion	string	Version of the software's component
WinAsIsCAPs	string	List of non-modified Central Access Policies
WinAttributeLDAPDisplayName	string	Attributes of executable file being processed
WinAttributes	string	Attributes of the object or event
WinAttributeSyntaxOID	string	File attributes of an object being processed
WinAttributeValue	string	Attributes of executable file being processed
WinAuditPolicyChanges	string	Security parameters from windows event log events
WinAuditSourceName	string	Name of the object being processed
WinAuditStatusCode	string	Event audit status code
WinAuthenticationLevel	string	Authentication level for the request or connection
WinAuthenticationPackageName	string	Name of the object being processed
WinAuthenticationProvider	string	Authentication provider for the event
WinAuthenticationServer	string	Server that performed the authentication
WinAuthenticationService	string	Authentication service used in the event
WinAuthenticationType	string	Authentication type for the event
WinBackupType	string	Backup type
WinBcdCCID	string	Smart card device settings and properties
WinBMaxSlotIndex	string	Smart card device settings and properties
WinBNumClockSupported	string	Smart card device settings and properties
WinBNumDataRateSupported	string	Smart card device settings and properties
WinBootMode	string	User sessions parameters from windows event log events
WinBuildVersion	string	Version of the statistics being sent
WinBVoltageSupport	string	Smart card device settings and properties
WinCalledStationID	string	Called station identifier in authentication
WinCallerProcessId	string	Parent process system id (pid)
WinCallerProcessName	string	Name of the object being processed
WinCallingStationID	string	Calling station identifier in authentication
WinCalloutKey	string	Key of the called function or rule
WinCalloutName	string	Name of the called function or rule
WinCategoryId	string	Operations parameters from windows event log events

Field	Type	Description
WinCategoryName	string	Category of the event or threat
WinCertificateTemplate	string	Certificate template used in the event
WinCertIssuerName	string	Certificate issuer name
WinCertSerialNumber	string	Certificate serial number
WinCertThumbprint	string	Digital certificate thumbprint of the scanned object and hashing algorithm
WinChangeType	string	Type of change made
WinClassId	string	Object type code
WinClassName	string	Name of the object being processed
WinClientAddress	string	Local ip address
WinClientIPAddress	string	Client IP address initiating the event
WinClientName	string	Id of the account under which the controlled process was started
WinClientProcessId	string	Client process identifier
WinClientProcessStartKey	string	The unique key of the process that created the task
WinCommandLine	string	Command line of the launched process
WinCompatibleIds	string	Windows event log events additional parameters
WinComputerAccountChange	string	Operations parameters from windows event log events
WinComputerName	string	Name of the computer where the event occurred
WinConditions	string	Conditions related to the event or filter
WinConfigAccessPolicy	string	Configuration access policy
WinConfiguredNames	string	Configured names of resources or objects
WinContextInfo	string	Tasks parameters recorded in the windows event log events
WinCountOfCredentialsReturned	string	Number of returned credentials
WinCrashOnAuditFailValue	string	Value of crash-on-audit-fail setting
WinCredType	string	Type of credentials
WinCurrentEngineVersion	string	Current version of the security engine
WinCurrentSignatureVersion	string	Current signature version
WinCVEID	string	Windows event log events additional parameters
WinDCDNSName	string	Domain Controller DNS name
WinDCOMorRPC	string	DCOM or RPC protocol used
WinDeletedCAPs	string	List of deleted Central Access Policies
WinDestAddress	string	Destination address for network event
WinDestinationDRA	string	Network connections parameters from windows event log events
WinDestPort	string	Destination port for network event
WinDetectionID	string	Identifier of the detected threat
WinDetectionSource	string	Source of the threat detection
WinDetectionSourceIndex	string	Index of the detection source
WinDetectionTime	string	Time when the threat was detected
WinDetectionType	string	Type of threat detection
WinDetectionTypeIndex	string	Index of the detection type

Field	Type	Description
WinDetectionUser	string	User under whom the threat was detected
WinDeviceClaims	string	Devices involved in authentication
WinDeviceDescription	string	Description of an object being processed as defined in the object properties
WinDeviceId	string	Device id
WinDirection	string	Direction of network traffic
WinDisabledPrivilegeList	string	List of disabled privileges
WinDisableIntegrityChecks	string	Flag to disable integrity checks
WinDisplayName	string	Id of the account under which the controlled process was started
WinDisposition	string	Result of processing event or object
WinDnsHostName	string	User device name
WinDnsName	string	User device name
WinDomain	string	Domain associated with the event or object
WinDomainBehaviorVersion	string	Active directory objects parameters
WinDomainName	string	Name of the domain used to authenticate the owner of the logon session
WinDomainPolicyChanged	string	Active directory objects parameters
WinDomainSid	string	User account security identifier (sid)
WinDSName	string	Name of the object being processed
WinDSType	string	Type of the object detected by the software
WinDummy	string	Windows event log events additional parameters
WinDuration	string	Duration of the event
WinDwDataRate	string	Smart card device settings and properties
WinDwDefaultClock	string	Smart card device settings and properties
WinDwFeatures	string	Smart card device settings and properties
WinDwMaxDataRate	string	Smart card device settings and properties
WinDwMaxIFSD	string	Smart card device settings and properties
WinDwMaximumClock	string	Smart card device settings and properties
WinDwMechanical	string	Smart card device settings and properties
WinDwProtocols	string	Smart card device settings and properties
WinDwSyncProtocols	string	Smart card device settings and properties
WinEAPErrorCode	string	EAP error code
WinEAPReasonCode	string	EAP reason code
WinEapRootCauseString	string	Description of the root cause of the EAP error
WinEAPType	string	Type of EAP authentication
WinEffectiveState	string	Current state of the object or system
WinElevatedToken	string	Security parameters from windows event log events
WinEnabledPrivilegeList	string	List of enabled privileges
WinEndUSN	string	End change identifier in the USN journal
WinEngineVersion	string	Engine version at the time of the event
WinEntryType	string	Type of the object detected by the software

Field	Type	Description
WinErrorCode	string	Operations parameters from windows event log events
WinErrorDescription	string	Description of the error occurred during the event
WinEventCount	string	Number of event repetitions
WinEventCountTotal	string	Total number of related events
WinEventId	string	Event identifier
WinEventIdx	string	Event index in the sequence
WinEventSourceId	string	Object type code
WinExecutionID	string	Identifier of the process that generated the event
WinExecutionName	string	Name of the process that generated the event
WinExecutionStatus	string	Execution status of the action
WinExecutionStatusIndex	string	Index of the execution status
WinExeFileName	string	Name of the object being processed
WinFailureId	string	Failure identifier
WinFailurePoint	string	Point where failure occurred
WinFailureReason	string	Reason for moving the file to the category
WinFileName	string	Name of the object being processed
WinFilterId	string	Filter identifier
WinFilterKey	string	Filter key
WinFilterName	string	Filter name
WinFilterRTID	string	Network connections parameters from windows event log events
WinFilterType	string	Filter type
WinFlags	string	Operations parameters from windows event log events
WinFlightSigning	string	Flag for flight build signing
WinForceLogoff	string	Active directory objects parameters
WinForestRoot	string	Id of the account under which the controlled process was started
WinForestRootSid	string	User account security identifier (sid)
WinFQDN	string	The fully qualified domain name (FQDN) of the host or computer from which the event was initiated
WinFullyQualifiedSubjectMachineName	string	Fully qualified machine name of the subject
WinFullyQualifiedSubjectUserName	string	Fully qualified username of the subject
WinFWLink	string	Link to the page describing the threat
WinGPOList	string	Access rights for the object being processed
WinGroupDomainName	string	Group domain name
WinGroupMembership	string	Group membership
WinGroupName	string	Group name
WinGroupPolicyApplied	string	Applied group policy
WinGroupSid	string	Group SID
WinGroupTypeChange	string	Information about user group
WinHandleId	string	Object type code
WinHangType	string	Type of the object detected by the software

Field	Type	Description
WinHardwareIds	string	Hardware identifiers
WinHomeDirectory	string	Path to the object being processed
WinHomePath	string	Path to the object being processed
WinHypervisorDebug	string	Hypervisor debug flag
WinHypervisorLaunchType	string	Hypervisor launch type
WinHypervisorLoadOptions	string	Hypervisor load options
WinIdentity	string	User or device identifier
WinImagePath	string	Path to the object being processed
WinImpersonationLevel	string	Security parameters from windows event log events
WinInitiatorCookie	string	Initiator operation cookie
WinInterfaceName	string	Network interface name
WinIntfGuid	string	Network interface GUID
WinIpAddress	string	Local ip address
WinIpAddresses	string	IP addresses associated with the object
WinIpPort	string	Port numbers of the source and the destination of the network connection being processed
WinKerberosPolicyChange	string	Security parameters from windows event log events
WinKernelDebug	string	Kernel debug flag
WinKeyFilePath	string	Key file path
WinKeyLength	string	Active directory objects parameters
WinKeyModName	string	Keying module name
WinKeyName	string	Cryptographic key name
WinKeyType	string	Cryptographic key type
WinLayerId	string	Filter layer identifier
WinLayerKey	string	Filter layer key
WinLayerName	string	Network connections parameters from windows event log events
WinLayerRTID	string	Network connections parameters from windows event log events
WinLinkName	string	Name of the object being processed
WinLmPackageName	string	Name of the object being processed
WinLoadOptions	string	Load options
WinLocalAddress	string	Local address in network event
WinLocalKeyModPort	string	Local key module port
WinLocalMac	string	Local MAC address
WinLocalMMPrincipalName	string	Local security principal that was used during the IPsec Main Mode authentication process
WinLocationInformation	string	User sessions parameters from windows event log events
WinLockoutDuration	string	Active directory objects parameters
WinLockoutObservationWindow	string	Active directory objects parameters
WinLockoutThreshold	string	Active directory objects parameters
WinLogDroppedPacketsEnabled	string	Flag indicating whether logging of dropped network packets is enabled

Field	Type	Description
WinLoggingResult	string	Logging result
WinLogonGuid	string	Information about user group
WinLogonHours	string	Active directory objects parameters
WinLogonID	string	User account security identifier (sid)
WinLogonProcessName	string	Object name derived from the windows event log
WinLogonType	string	Logon session type
WinLogSuccessfulConnectionsEnabled	string	Flag indicating whether logging of successful network connections is enabled
WinMachineAccountQuota	string	Active directory objects parameters
WinMachineInventory	string	Machine inventory
WinMajorVersion	string	Version of the statistics being sent
WinMandatoryLabel	string	Mandatory security label of the process
WinMasterKeyId	string	Master key identifier for data encryption
WinMaxPasswordAge	string	Active directory objects parameters
WinMemberName	string	Id of the account under which the controlled process was started
WinMemberSid	string	User account security identifier (sid)
WinMessageNumber	string	Windows event log events additional parameters
WinMessageTotal	string	Windows event log events additional parameters
WinMinorVersion	string	Version of the statistics being sent
WinMinPasswordAge	string	Active directory objects parameters
WinMinPasswordLength	string	Active directory objects parameters
WinMixedDomainMode	string	Active directory objects parameters
WinMMAuthMethod	string	The authentication method used during IPSec main mode negotiation
WinMMFilterID	string	Identifier that Windows assigns to a specific IPSec Main Mode filter
WinMMImpersonationState	string	Field shows whether impersonation was used during the IPSec Main Mode authentication process
WinModifiedCAPs	string	List of modified Central Access Policies
WinModifiedObjectProperties	string	Description of an object being processed as defined in the object properties
WinMulticastFlowsEnabled	string	Multicast flows enabled flag
WinNamingContext	string	Network connections parameters from windows event log events
WinNASIdentifier	string	Network Access Server (NAS) identifier
WinNASIPv4Address	string	Network Access Server (NAS) IPv4 address
WinNASIPv6Address	string	Network Access Server (NAS) IPv6 address
WinNASPort	string	Network Access Server (NAS) port for the connection
WinNASPortType	string	Network Access Server (NAS) port type
WinNetbiosName	string	Name of the object being processed
WinNetworkPolicyName	string	Applied network policy name
WinNewMaxUsers	string	Distributed file system resources properties
WinNewObjectDN	string	Object distinguished name

Field	Type	Description
WinNewProcessId	string	New process ID
WinNewProcessName	string	New process name
WinNewRemark	string	Distributed file system resources properties
WinNewSd	string	Security parameters from windows event log events
WinNewShareFlags	string	Distributed file system resources properties
WinNewState	string	New state of the object or system
WinNewTargetUserName	string	Id of the account under which the controlled process was started
WinNewTime	string	Event time
WinNewUacValue	string	Properties of user account control component
WinNewValue	string	File attributes of an object being processed
WinNewValueType	string	Type of the object detected by the software
WinNotificationPackageName	string	Name of the object being processed
WinObjectClass	string	Bit mask of application categories in the application update component
WinObjectCollectionName	string	Name of the object being processed
WinObjectDN	string	Object distinguished name
WinObjectGUID	string	Information about user group
WinObjectIdentifyingProperties	string	Windows event log events additional parameters
WinObjectName	string	Object name derived from the windows event log
WinObjectProperties	string	Tasks parameters recorded in the windows event log events
WinObjectServer	string	Tasks parameters recorded in the windows event log events
WinObjectType	string	Type of the object detected by the software
WinObjectValueName	string	Name of the object being processed
WinOemInformation	string	Description of an object being processed as defined in the object properties
WinOldMaxUsers	string	Distributed file system resources properties
WinOldObjectDN	string	Object distinguished name
WinOldRemark	string	Distributed file system resources properties
WinOldSd	string	Security parameters from windows event log events
WinOldShareFlags	string	Distributed file system resources properties
WinOldTargetUserName	string	Id of the account under which the controlled process was started
WinOldUacValue	string	Properties of user account control component
WinOldValue	string	File attributes of an object being processed
WinOldValueType	string	Type of the object detected by the software
WinOpCorrelationID	string	Object type code
WinOperation	string	Type of operation performed
WinOperationId	string	Operations parameters from windows event log events
WinOperationMode	string	Operation execution mode
WinOperationType	string	Type of the object detected by the software
WinOptions	string	Operations parameters from windows event log events
WinOriginID	string	Identifier of the object origin

Field	Type	Description
WinOriginName	string	Name of the object origin
WinPackage	string	Name or identifier of the package related to the event
WinPackageFullName	string	Name of the object being processed
WinPackageName	string	Name of the object being processed
WinPackageRelativeAppId	string	Object type code
WinPackageSid	string	Package SID related to the event
WinParam1	string	Query string
WinParam2	string	Query string
WinParam3	string	Query string
WinParam4	string	Query string
WinParam5	string	Query string
WinParentProcessId	string	The ID of the parent process that launched the current process when the task was created
WinParentProcessName	string	Parent process name
WinPasswordHistoryLength	string	Active directory objects parameters
WinPasswordLastSet	string	Active directory objects parameters
WinPasswordProperties	string	Description of an object being processed as defined in the object properties
WinPath	string	Path to the object being processed
WinPathFound	string	Path to the detected file or object
WinPayload	string	File attributes of an object being processed
WinPeerMac	string	Peer device MAC address
WinPostCleanStatus	string	Status after threat remediation
WinPreAuthType	string	Type of the object detected by the software
WinPreExecutionStatus	string	Status before executing the action
WinPreviousEngineVersion	string	Previous version of the security engine
WinPreviousSignatureVersion	string	Previous signature version
WinPreviousTime	string	Event time
WinPrimaryGroupId	string	Information about user group
WinPrivilegeList	string	Access rights for the object being processed
WinProcessCreationTime	string	Process creation time
WinProcessId	string	Process system id (pid)
WinProcessName	string	Name of the object being processed
WinProductName	string	Name of the security product that generated the event
WinProductVersion	string	Version of the security product
WinProfile	string	User or system profile name/ID
WinProfileChanged	string	Operations parameters from windows event log events
WinProfilePath	string	Path to the object being processed
WinProfiles	string	List of applied profiles
WinProfileUsed	string	Profile used in the event
WinProperties	string	Description of an object being processed as defined in the object properties

Field	Type	Description
WinPropertyName	string	Property name of the object
WinProtocol	string	Network connections parameters from windows event log events
WinProviderKey	string	Provider key for filter or operation
WinProviderName	string	Provider name for operation or event
WinProxyPolicyName	string	Proxy policy name
WinPuaCount	string	Windows event log events additional parameters
WinPuaPolicyId	string	Object type code
WinQfeVersion	string	Version of the statistics being sent
WinQuarantineSessionIdentifier	string	Quarantine session identifier
WinQuarantineState	string	Object state in quarantine
WinReadOperation	string	Flag for read operation
WinReason	string	Reason for the event or action
WinReasonCode	string	Reason code for the event
WinReasonForRejection	string	Reason for rejection of the operation
WinReasonText	string	Text description of the reason for the event
WinRecoveryKeyId	string	Recovery key identifier
WinRecoveryReason	string	Reason for using recovery key
WinRecoveryServer	string	Data recovery server
WinRelativeTargetName	string	Name of the object being processed
WinRemediationUser	string	User who initiated the remediation
WinRemoteAddress	string	Remote address of the connection or operation
WinRemoteAdminEnabled	string	Remote administration enabled flag
WinRemoteEventLogging	string	Remote event logging flag
WinRemoteKeyModPort	string	Remote key module port
WinRemoteMachineID	string	Remote machine identifier
WinRemoteMMPrincipalName	string	Remote security principal that was used during the IPsec Main Mode authentication process
WinRemoteUserID	string	Remote user identifier
WinReplicationEvent	string	Replication event flag
WinReportId	string	Object type code
WinRequestClientInfo	string	Client information sending the request
WinRequestCSPProvider	string	Requested CSP provider in the event
WinRequester	string	Requester
WinRequestId	string	Request ID
WinRequestOSVersion	string	OS version of the requesting client
WinResource	string	Resource or object involved in the operation
WinResourceAttributes	string	File attributes of an object being processed
WinResourceManager	string	Resource manager responsible for the object
WinResponderCookie	string	Responder operation cookie
WinRestrictedAdminMode	string	Security parameters from windows event log events
WinRestrictedSidCount	string	Number of restricted SIDs for the operation

Field	Type	Description
WinReturnCode	string	Return code of the operation result
WinRole	string	Role of the subject in the operation
WinRpcCallClientLocality	string	The RPC client locality that initiated the event, indicating whether the call was local or remote
WinRuleAttr	string	Security rule or filter attribute
WinRuleId	string	Object type code
WinRuleName	string	Name of the object being processed
WinRunspaceId	string	PowerShell runspace ID
WinSamAccountName	string	Id of the account under which the controlled process was started
WinScanID	string	Identifier of the scan
WinScanParameters	string	Scan parameters
WinScanParametersIndex	string	Index of the scan parameters
WinScanResources	string	Resources that were scanned
WinScanTimeHours	string	Hours taken for the scan
WinScanTimeMinutes	string	Minutes taken for the scan
WinScanTimeSeconds	string	Seconds taken for the scan
WinScanType	string	Type of scan
WinScanTypeIndex	string	Index of the scan type
WinSchema	string	Schema of the object or event
WinSchemaFriendlyName	string	Friendly name of the schema
WinScriptBlockId	string	Object type code
WinScriptBlockText	string	User sessions parameters from windows event log events
WinScriptPath	string	Path to the object being processed
WinSecurityDescriptor	string	Security descriptor of the object
WinSecurityPackageName	string	Name of the object being processed
WinSerialNumber	string	Serial number of the certificate or security object
WinServerNames	string	List of server names involved in the operation
WinService	string	Name of the service or service component associated with the event
WinServiceAccount	string	Id of the account under which the controlled process was started
WinServiceFileName	string	Name of the object being processed
WinServiceName	string	Object name derived from the windows event log
WinServicePrincipalNames	string	Name of the object being processed
WinServiceSid	string	User account security identifier (sid)
WinServiceStartType	string	Start type of the windows service
WinServiceType	string	Type of the object detected by the software
WinServiceVersion	string	Version of the statistics being sent
WinSessionId	string	Unique id of the logon session
WinSessionName	string	User sessions parameters from windows event log events
WinSettingType	string	Type of the object detected by the software
WinSettingValue	string	File attributes of an object being processed

Field	Type	Description
WinSeverityID	string	Identifier of the threat severity level
WinSeverityName	string	Name of the threat severity level
WinShareLocalPath	string	Path to the object being processed
WinShareName	string	Id of the account under which the controlled process was started
WinSID	string	Security Identifier (SID) related to a user or object
WinSidFilteringEnabled	string	Security parameters from windows event log events
WinSidHistory	string	User account security identifier (sid)
WinSidList	string	User account security identifier (sid)
WinSignatureType	string	Type of threat signature
WinSignatureTypeIndex	string	Index of the signature type
WinSignatureVersion	string	Signature version at the time of the event
WinSourceAddr	string	Network connections parameters from windows event log events
WinSourceAddress	string	Local ip address
WinSourceDRA	string	Network connections parameters from windows event log events
WinSourceHandleId	string	Handle ID of the source object
WinSourceID	string	Identifier of the event or action source
WinSourceName	string	Name of the event or action source
WinSourcePort	string	Port numbers of the source and the destination of the network connection being processed
WinSourceProcessId	string	Source process ID
WinSourceSid	string	SID of the subject that initiated the operation
WinSourceUserName	string	Source username
WinSpnName	string	SPN (Service Principal Name) used for authentication
WinSSID	string	Wireless network identifier (SSID)
WinStagingReason	string	Reason for staging the object for audit
WinStartTime	string	Event time
WinStartType	string	Type of the object detected by the software
WinStartUSN	string	Network connections parameters from windows event log events
WinState	string	State of the object or system at the time of the event
WinStatus	string	Description of an object being processed as defined in the object properties
WinStatusCode	string	Operations parameters from windows event log events
WinStatusDescription	string	Description of the current status of the event or action
WinStopTime	string	Event time
WinSubcategoryGuid	string	Information about user group
WinSubcategoryId	string	Object type code
WinSubject	string	Information about the subject that initiated the action
WinSubjectAlternativeName	string	Certificate SAN (Subject Alternative Name) field containing alternate identities
WinSubjectDomainName	string	Name of the domain used to authenticate the owner of the logon session

Field	Type	Description
WinSubjectKeyIdentifier	string	Certificate Subject Key Identifier
WinSubjectLogonId	string	User account security identifier (sid)
WinSubjectMachineName	string	Name of the machine on whose behalf the action was performed
WinSubjectMachineSID	string	Machine SID for the system that performed the action
WinSubjectUserDomainName	string	Name of the domain used to authenticate the owner of the logon session
WinSubjectUserName	string	Id of the account under which the controlled process was started
WinSubjectUserSid	string	User account security identifier (sid)
WinSubStatus	string	Operations parameters from windows event log events
WinTargetDomainName	string	Name of the domain used to authenticate the owner of the logon session
WinTargetHandleId	string	Handle ID of the target object
WinTargetInfo	string	Description of an object being processed as defined in the object properties
WinTargetLinkedLogonId	string	User account security identifier (sid)
WinTargetLogonGuid	string	Information about user group
WinTargetLogonId	string	User account security identifier (sid)
WinTargetName	string	Name of the target object or resource
WinTargetOutboundDomainName	string	Name of the domain used to authenticate the owner of the logon session
WinTargetOutboundUserName	string	Id of the account under which the controlled process was started
WinTargetProcessId	string	Process system id (pid)
WinTargetProcessName	string	Name of the object being processed
WinTargetServer	string	Target server of the event
WinTargetServerName	string	User device name
WinTargetSid	string	User account security identifier (sid)
WinTargetState	string	State of the target object after the operation
WinTargetUserDomain	string	Domain of the target user
WinTargetUserName	string	Id of the account under which the controlled process was started
WinTargetUserSid	string	User account security identifier (sid)
WinTaskContent	string	Original content of the scheduled task associated with the event
WinTaskContentActionsContext	string	Operating system task scheduler tasks parameters
WinTaskContentActionsExecCommand	string	Command line arguments for the process
WinTaskContentPrincipalsPrincipalid	string	User account security identifier (sid)
WinTaskContentPrincipalsPrincipalLogonType	string	Logon session type
WinTaskContentPrincipalsPrincipalRunLevel	string	Operating system task scheduler tasks parameters
WinTaskContentPrincipalsPrincipalUserId	string	User account security identifier (sid)
WinTaskContentRegistrationInfoAuthor	string	Id of the account under which the controlled process was started
WinTaskContentRegistrationInfoDate	string	Date of event
WinTaskContentSettingsAllowHardTerminate	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsAllowStartOnDemand	string	Operating system task scheduler tasks parameters

Field	Type	Description
WinTaskContentSettingsDisallowStartIfOnBatteries	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsEnabled	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsExecutionTimeLimit	string	Event time
WinTaskContentSettingsHidden	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsIdleSettingsRestartOnIdle	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsIdleSettingsStopOnIdleEnd	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsMultipleInstancesPolicy	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsPriority	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsRunOnlyIfIdle	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsRunOnlyIfNetworkAvailable	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsStartWhenAvailable	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsStopIfGoingOnBatteries	string	Operating system task scheduler tasks parameters
WinTaskContentSettingsWakeToRun	string	Operating system task scheduler tasks parameters
WinTaskContentTriggers	string	Tasks parameters recorded in the windows event log events
WinTaskContentversion	string	Version of the task in its original content
WinTaskContentNew	string	Updated content of the scheduled task
WinTaskContentNewActionsContext	string	Operating system task scheduler tasks parameters
WinTaskContentNewActionsExecCommand	string	Command line arguments for the process
WinTaskContentNewPrincipalsPrincipalid	string	User account security identifier (sid)
WinTaskContentNewPrincipalsPrincipalLogonType	string	Logon session type
WinTaskContentNewPrincipalsPrincipalRunLevel	string	Operating system task scheduler tasks parameters
WinTaskContentNewPrincipalsPrincipalUserId	string	User account security identifier (sid)
WinTaskContentNewRegistrationInfoAuthor	string	Id of the account under which the controlled process was started
WinTaskContentNewRegistrationInfoDate	string	Date of event
WinTaskContentNewSettingsAllowHardTerminate	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsAllowStartOnDemand	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsDisallowStartIfOnBatteries	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsEnabled	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsExecutionTimeLimit	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsHidden	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsIdleSettingsRestartOnIdle	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsIdleSettingsStopOnIdleEnd	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsMultipleInstancesPolicy	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsPriority	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsRunOnlyIfIdle	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsRunOnlyIfNetworkAvailable	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsStartWhenAvailable	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsStopIfGoingOnBatteries	string	Operating system task scheduler tasks parameters
WinTaskContentNewSettingsWakeToRun	string	Operating system task scheduler tasks parameters
WinTaskContentNewTriggers	string	Tasks parameters recorded in the windows event log events

Field	Type	Description
WinTaskContentNewversion	string	Version of the task in its updated content
WinTaskName	string	Name of the object being processed
WinTdoAttributes	string	File attributes of an object being processed
WinTdoDirection	string	Network connections parameters from windows event log events
WinTdoType	string	Type of the object detected by the software
WinTemplateContent	string	Content of the certificate template
WinTemplateDSObjectFQDN	string	FQDN of the template object in Active Directory
WinTemplateInternalName	string	Internal name of the certificate template
WinTemplateOID	string	OID (Object Identifier) of the certificate template
WinTemplateSchemaVersion	string	Schema version of the certificate template
WinTemplateVersion	string	Version of the certificate template
WinTerminationTime	string	Event time
WinTestSigning	string	Flag indicating whether Test Signing mode is enabled on the system
WinThreatID	string	Identifier of the threat
WinThreatName	string	Name of the threat
WinTicketEncryptionType	string	Type of the object detected by the software
WinTicketOptions	string	Active directory objects parameters
WinTokenElevationType	string	Security token elevation type
WinTopLevelName	string	Name of the object being processed
WinTransactionId	string	Object type code
WinTransitionsToOn	string	Transition of the object or component to the 'on' state
WinTransmittedServices	string	Operations parameters from windows event log events
WinTreeDelete	string	Windows event log events additional parameters
WinType	string	Type of the event or object
WinTypeID	string	Identifier of the event or threat type
WinTypeName	string	Name of the event or threat type
WinUnused	string	Reserved or unused field
WinUnused2	string	Additional reserved field not used by the event
WinUnused3	string	Additional reserved field not used by the event
WinUnused4	string	Additional reserved field not used by the event
WinUnused5	string	Additional reserved field not used by the event
WinUnused6	string	Additional reserved field not used by the event
WinUpdateType	string	Type of product or signature update
WinUpdateTypeIndex	string	Index of the update type
WinUser	string	Username associated with the event
WinUserAccountControl	string	Properties of user account control component
WinUserClaims	string	List of user claims for new logon session
WinUserData	string	User sessions parameters from windows event log events
WinUserName	string	Username

Field	Type	Description
WinUserParameters	string	Id of the account under which the controlled process was started
WinUserPrincipalName	string	Id of the account under which the controlled process was started
WinUserSid	string	SID of the user associated with the event
WinUserUPN	string	User Principal Name (UPN) of the user
WinUserWorkstations	string	User device name
WinVendorIds	string	User sessions parameters from windows event log events
WinVirtualAccount	string	Id of the account under which the controlled process was started
WinVsmLaunchType	string	Launch type of Virtual Secure Mode (VSM)
WinWakeFromState	string	Power state from which the system was awakened (sleep, hibernation, etc.)
WinWakeRequesterTypeAc	string	Wake requester type when running on AC power
WinWakeRequesterTypeDc	string	Wake requester type when running on DC power
WinWeight	string	Weight or priority associated with the event object
WinWorkstation	string	User device name
WinWorkstationName	string	User device name
WinProcessingErrorDataDataItemName	string	Name of the data item that caused a processing error
WinProcessingErrorDataErrorCode	string	Error code that occurred during event processing
WinProcessingErrorDataEventPayload	string	Original event payload that caused the error
WinSystemChannel	string	Windows Event Log channel from which the event was recorded
WinSystemComputer	string	Name of the computer that generated the event
WinSystemEventID	string	The event identifier that defines the type of event that occurred in the system
WinSystemEventIDQualifiers	string	Additional event qualifiers providing extended information about the event
WinSystemEventRecordID	string	A unique record identifier in the log
WinSystemExecutionProcessID	string	The process identifier (PID) associated with the event execution
WinSystemExecutionThreadID	string	The thread identifier (TID) that triggered the event
WinSystemKeywords	string	Event keywords used for categorizing and filtering events
WinSystemLevel	string	The event level indicating its severity or importance
WinSystemOpcode	string	The event operation code describing the type of action associated with the event
WinSystemProviderEventSourceName	string	The name of the source that generated the event
WinSystemProviderGuid	string	The globally unique identifier of the event provider
WinSystemProviderName	string	The name of the event provider
WinSystemSecurityUserID	string	The identifier of the user associated with the event
WinSystemTask	string	The task identifier associated with the event
WinSystemTimeCreatedSystemTime	string	The timestamp when the event was created
WinSystemVersion	string	The version of the event schema
WinAutoBackupBackupPath	string	The file or folder path where the automatic backup is stored
WinAutoBackupChannel	string	The channel or source used for automatic backup
WinEventProcessingFailureErrorCode	string	Error code encountered during event processing

Field	Type	Description
WinEventProcessingFailureEventID	string	The ID of the event that caused the processing failure
WinEventProcessingFailurePublisherID	string	Identifier for the application or service that was trying to log the event that failed
WinEventXMLAddress	string	Address associated with the event
WinEventXMLParam1	string	Additional event information field
WinEventXMLParam2	string	Additional event information field
WinEventXMLParam3	string	Additional event information field
WinEventXMLSessionID	string	The user session ID associated with the event
WinEventXMLSource	string	The source of the event
WinEventXMLTargetSession	string	The target session related to the event
WinEventXMLUser	string	The user associated with the event
WinFileIsFull	string	An indicator that the log is full and can no longer accept new entries.
WinLogFileClearedSubjectDomainName	string	The domain of the user who cleared the event log
WinLogFileClearedSubjectLogonId	string	The logon ID of the user who cleared the event log
WinLogFileClearedSubjectUserName	string	The username of the user who cleared the event log
WinLogFileClearedSubjectUserSid	string	The SID of the user who cleared the event log
WinOperationClientFailureClientMachine	string	The name of the client machine where the operation failure occurred
WinOperationClientFailureClientProcessId	string	The process ID of the client that caused the operation failure
WinOperationClientFailureComponent	string	The system component where the operation failure occurred
WinOperationClientFailureId	string	The unique identifier of the operation failure
WinOperationClientFailureOperation	string	The name or description of the operation that failed
WinOperationClientFailurePossibleCause	string	The possible cause of the operation failure
WinOperationClientFailureResultCode	string	The result code of the operation indicating success or failure reason
WinOperationClientFailureUser	string	The user who initiated the operation that failed
WinOperationStartedOperationalCode	string	The code of the started operation
WinOperationStartedOperationalHostProcess	string	The host process that initiated the operation
WinOperationStartedOperationalProcessID	string	The process ID associated with the started operation
WinOperationStartedOperationalProviderName	string	The name of the provider or event source
WinOperationStartedOperationalProviderPath	string	The path to the provider or event source
WinServiceShutdown	string	A Windows service has been stopped

Blocked Document

This event type is supported on Windows and Linux operating systems.

Blocked Document event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack

Field	Type	Description
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id

Field	Type	Description
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file

Field	Type	Description
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file

Field	Type	Description
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded

Field	Type	Description
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Applock

This event type is supported on Windows, Linux, and macOS operating systems.

Applock event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack

Field	Type	Description
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessCalltrace	string	Stack trace of the process
ProcessFileLnkName	string	Name of the LNK file that was used for the process execution (without the path)
ProcessFileLnkPath	string	Path to the LNK file that was used for the process execution (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was used for the process execution
ProcessFileLnkCmdline	string	Command line of the LNK file for the process execution
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS

Field	Type	Description
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS

Field	Type	Description
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file

Field	Type	Description
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)

Field	Type	Description
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process

Field	Type	Description
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process

Field	Type	Description
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Process Terminated

This event type is supported on Windows, Linux, and macOS operating systems.

Process Terminated event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version

Field	Type	Description
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessCalltrace	string	Stack trace of the process
ProcessFileLnkName	string	Name of the LNK file that was used for the process execution (without the path)
ProcessFileLnkPath	string	Path to the LNK file that was used for the process execution (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was used for the process execution
ProcessFileLnkCmdline	string	Command line of the LNK file for the process execution
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file

Field	Type	Description
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS

Field	Type	Description
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id

Field	Type	Description
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file

Field	Type	Description
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
ProcessTerminationtime	int	The time the process was completed

Device

This event type is supported on Windows operating systems.

Device event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventVersion	string	Event schema version
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
DeviceId	string	Device id
DeviceName	string	Device name
DevicePath	string	Device path
DeviceOperationtype	string	Device operation type (e.g. 1 - connect)
DeviceType	string	Device type
DeviceSystemflag	string	Flag if device is system

File System Mount

This event type is supported on Windows and Linux operating systems.

File System Mount event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version

Field	Type	Description
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
VolumeName	string	Disk volume name
VolumePath	string	Disk volume mount point
DeviceClass	string	GUID for device class according to official specification
DeviceId	string	Device id
DeviceBustype	string	Bus type used by connected device
DeviceType	string	Device type
DeviceHotpluggable	string	Is device hot pluggable or not (true or false)
DeviceSerialnumber	string	Manufacturer device ID
VolumeFilesystemtype	string	File system type of disk volume
DeviceSpaceTotal	uint	Total disk capacity (in Mb)
DeviceSpaceUsed	uint	Used space on the disk (in Mb)
DeviceSpaceFree	uint	Available free space on the disk (in Mb)

File System Unmount

This event type is supported on Windows and Linux operating systems.

File System Unmount event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
VolumeName	string	Disk volume name
VolumePath	string	Disk volume mount point
DeviceClass	string	GUID for device class according to official specification
DeviceId	string	Device id
DeviceBustype	string	Bus type used by connected device
DeviceType	string	Device type

Field	Type	Description
DeviceHotpluggable	string	Is device hot pluggable or not (true or false)
DeviceSerialnumber	string	Manufacturer device ID
VolumeFilesystemtype	string	File system type of disk volume
DeviceSpaceTotal	uint	Total disk capacity (in Mb)
DeviceSpaceUsed	uint	Used space on the disk (in Mb)
DeviceSpaceFree	uint	Available free space on the disk (in Mb)

Bits Job Create

This event type is supported on Windows operating systems.

Bits Job Create event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name

Field	Type	Description
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
BitsjobId	string	Identifier of job (GUID)
BitsjobName	string	The name of job
BitsjobOwnerUserName	string	Username of job owner
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the file that was executed
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID

Field	Type	Description
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file

Field	Type	Description
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Creation flags of grand-parent process
GrandParentProcessFileLnkName	string	Name without the path of the LNK file that started the grand-parent process
GrandParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the grand-parent process
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Bits Job Complete

This event type is supported on Windows operating systems.

Bits Job Complete event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version

Field	Type	Description
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
BitsjobId	string	Identifier of job (GUID)
BitsjobName	string	The name of job
BitsjobOwnerUserName	string	Username of job owner
BitsjobUserName	string	Username who performs the operation
BitsjobFilestotal	int	Total count of files in job
BitsjobBytestransferred	int	Total count of transferred bytes in job
BitsjobPeerBytestransferred	int	The count of bytes transferred via p2p

Bits Job Start

This event type is supported on Windows operating systems.

Bits Job Start event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
BitsjobId	string	Identifier of job (GUID)
BitsjobName	string	The name of job
BitsjobTransferid	string	Unique identifier of file transferring in job (GUID)
BitsjobRemotename	string	Data source URL
BitsjobDatasourceCreationtime	int	Creation time of data source
BitsjobDatasourceSize	int	Size of data source
BitsjobBytestotal	int	Total count of files in job
BitsjobBytestransferred	int	Total count of transferred bytes in job
BitsjobPeerHostname	int	Hostname of p2p data source
BitsjobPeerBytestransferred	int	The count of bytes transferred via p2p

Bits Job Stop

This event type is supported on Windows operating systems.

Bits Job Stop event fields

Field	Type	Description
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
BitsjobId	string	Identifier of job (GUID)
BitsjobName	string	The name of job
BitsjobTransferid	string	Unique identifier of file transferring in job (GUID)
BitsjobRemotename	string	Data source URL
BitsjobDatasourceCreationtime	int	Creation time of data source
BitsjobDatasourceSize	int	Size of data source

Field	Type	Description
BitsjobBytestotal	int	Total count of files in job
BitsjobBytestransferred	int	Total count of transferred bytes in job
BitsjobResultStatuscode	int	The result of data transferring operation via BITS
BitsjobResultAdditionalinfo	int	Additional information about the result of data transferring via BITS
BitsjobPeerHostname	string	Hostname of p2p data source
BitsjobPeerContextinfo	int	Identifying information of p2p data source
BitsjobPeerProtocolflags	int	Configuration flags that determine if the computer serves content to peers and can download content from peers
BitsjobPeerBytestransferred	int	The count of bytes transferred via p2p
BitsjobProxy	string	Proxy server that was used in transferring data
BitsjobBandwidthLimit	int	The maximum speed that a BITS job can use for downloading or uploading data
BitsjobBandwidthIgnoreonlan	int	The parameter determines whether to ignore speed limits if the source or destination is on a local network

Bits Job Add File

This event type is supported on Windows operating systems.

Bits Job Add File event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event

Field	Type	Description
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
BitsjobId	string	Identifier of job (GUID)
BitsjobName	string	The name of job
BitsjobOwnerUserName	string	Username of job owner
BitsjobUserName	string	Username who performs the operation
BitsjobFilestotal	int	Total count of files in job
BitsjobRemotename	string	Data source URL
BitsjobLocalname	string	Local path to file for saving job data
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file

Field	Type	Description
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process

Field	Type	Description
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process

Field	Type	Description
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Bits Job Error

This event type is supported on Windows operating systems.

Bits Job Error event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host

Field	Type	Description
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
BitsjobErrorid	int	Error Identifier
BitsjobId	string	Identifier of job (GUID)
BitsjobName	string	The name of job
BitsjobTransferid	string	Unique identifier of file transferring in job (GUID)
BitsjobRemotename	string	Data source URL
BitsjobDatasourceCreationtime	int	Creation time of data source
BitsjobDatasourceSize	int	Size of data source
BitsjobBytestotal	int	Total count of bytes in job
BitsjobBytestransferred	int	Total count of transferred bytes in job
BitsjobResultStatuscode	int	The result of data transferring operation via BITS
BitsjobResultAdditionalinfo	int	Additional information about the result of data transferring via BITS
BitsjobPeerHostname	string	Hostname of p2p data source
BitsjobPeerContextinfo	int	Identifying information of p2p data source
BitsjobPeerProtocolflags	int	Configuration flags that determine if the computer serves content to peers and can download content from peers.
BitsjobPeerBytestransferred	int	The count of bytes transferred via p2p
BitsjobProxy	string	Proxy server that was used in transferring data
BitsjobBandwidthLimit	int	The maximum speed that a BITS job can use for downloading or uploading data
BitsjobBandwidthIgnoreonlan	uint	The parameter determines whether to ignore speed limits if the source or destination is on a local network
BitsjobOwnerUserName	string	Username of job owner

Service Create

This event type is supported on Windows and Linux operating systems.

Service Create event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)

Field	Type	Description
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileAttr	int	The file system attributes
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file

Field	Type	Description
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ServiceTitle	string	Service name, unique identifier of service
ServiceType	string	Service type
ServiceStarttype	string	The type of service start
ServiceCmdline	string	Command line of started service
ServiceLoadordergroup	string	Load order group name of this service
ServiceTagid	uint	Unique tag value for this service in ServiceLoadOrderGroup
ServiceDependencies	string	Load order group names or service names that must start before this service
ServiceStartname	string	The account name from which the service will be launched
ServiceDisplayname	string	Displayed name of service
ServiceDeletedelayed	uint	Flag of delayed deletion. 1 - service will be deleted when it stops; 0 - service is deleted immediately. For "Service Create" event this field is optional.
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessProperties	string	List of properties of the account that runs process
ProcessIntegritylevel	int	Process integrity level
ProcessUserName	string	Domain and username who executed the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessUserName	string	Domain and username who executed parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessProperties	string	List of properties of the account which runs parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process

Service Modify

This event type is supported on Windows and Linux operating systems.

Service Modify event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name

Field	Type	Description
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileAttr	int	The file system attributes
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ServiceTitle	string	Service name, unique identifier of service
ServiceType	string	Service type
ServiceStarttype	string	The type of service start
ServiceCmdline	string	Command line of started service
ServiceLoadordergroup	string	Load order group name of this service
ServiceTagid	uint	Unique tag value for this service in ServiceLoadOrderGroup
ServiceDependencies	string	Load order group names or service names that must start before this service
ServiceStartname	string	The account name from which the service will be launched
ServiceDisplayname	string	Displayed name of service
ServiceOldType	string	Service type before modification
ServiceOldStarttype	string	The type of service start before modification
ServiceOldCmdline	string	The path to executable file before service modification
ServiceOldLoadordergroup	string	Load order group name of this service before modification
ServiceOldTagid	uint	Unique tag value for this service in ServiceLoadOrderGroup before modification
ServiceOldDependencies	string	Load order group names or service names that must be started prior to this service (before service modification)
ServiceOldStartname	string	The account name from which the service is launched before service modification
ServiceOldDisplayname	string	Displayed service name before service modification

Field	Type	Description
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessProperties	string	List of properties of the account that runs process
ProcessIntegritylevel	int	Process integrity level
ProcessUserName	string	Domain and username who executed the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessUserName	string	Domain and username who executed parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessProperties	string	List of properties of the account which runs parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process

Service Delete

This event type is supported on Windows and Linux operating systems.

Service Delete event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components

Field	Type	Description
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileType	string	Type of the file
FileAttr	int	The file system attributes
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ServiceTitle	string	Service name, unique identifier of service
ServiceType	string	Service type
ServiceStarttype	string	The type of service start
ServiceCmdline	string	Command line of started service
ServiceLoadordergroup	string	Load order group name of this service
ServiceTagid	uint	Unique tag value for this service in ServiceLoadOrderGroup
ServiceDependencies	string	Load order group names or service names that must start before this service
ServiceStartname	string	The account name from which the service will be launched
ServiceDisplayname	string	Displayed name of service
ServiceDeletedelayed	uint	Flag of delayed deletion. 1 - service will be deleted when it stops; 0 - service is deleted immediately. For "Service Create" event this field is optional.
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessProperties	string	List of properties of the account that runs process
ProcessIntegritylevel	int	Process integrity level
ProcessUserName	string	Domain and username who executed the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource

Field	Type	Description
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessUserName	string	Domain and username who executed parent process
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessProperties	string	List of properties of the account which runs parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process

Scheduled Task Create

This event type is supported on Windows and Linux operating systems.

Scheduled Task Create event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
TaskName	string	The name of task, unique identifier of service
TaskAuthorName	string	Username of task author
TaskUserName	string	Username from which the service will be launched
TaskUserId	int	Identifier of user from which the service will be launched. Only for Linux/Mac OS
TaskUserSid	string	SID of user from which the service will be launched. Only for Windows
TaskRunlevel	string	The level of privileges for task running
TaskCreationtime	int	Date and time of task creation
TaskCmdline	string	Actions performed in task
TaskEnabled	string	Flag whether the task is enabled
TaskHidden	string	Flag whether the task is hidden
TaskLocation	string	Path to the task object in the operating system
TaskSchedule	string	Schedule of task running
TaskSource	string	The source of task

Field	Type	Description
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name

Field	Type	Description
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field

Field	Type	Description
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process

Field	Type	Description
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process.
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process.
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed

Field	Type	Description
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process.
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process.
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process.
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment

Field	Type	Description
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Scheduled Task Modify

This event type is supported on Windows (Windows Vista, Windows Server 2008 and later) and Linux operating systems.

Scheduled Task Modify event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped

Field	Type	Description
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
TaskName	string	The name of task, unique identifier of service
TaskAuthorName	string	Username of task author
TaskUserName	string	Username from which the service will be launched
TaskUserSid	string	SID of user from which the service will be launched. Only for Windows
TaskRunlevel	string	The level of privileges for task running
TaskCreationtime	int	Date and time of task creation
TaskCmdline	string	Actions performed in task
TaskEnabled	string	Flag whether the task is enabled
TaskHidden	string	Flag whether the task is hidden
TaskLocation	string	Path to the task object in the operating system
TaskSchedule	string	Schedule of task running
TaskSource	string	The source of task
TaskOldUserName	string	Name of previous user from which the service was launched
TaskOldUserSid	string	SID of the previous user under which the task will be run
TaskOldEnabled	string	Previous value of flag whether the task is enabled
TaskOldHidden	string	Previous value of the flag indicating whether the task is hidden
TaskOldRunlevel	string	Previous level of privileges for task running
TaskOldCmdline	string	Previous actions performed in task

Field	Type	Description
TaskOldSchedule	string	Previous task execution schedule
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource

Field	Type	Description
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment

Field	Type	Description
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file

Field	Type	Description
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Scheduled Task Delete

This event type is supported on Windows and Linux operating systems.

Scheduled Task Delete event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack

Field	Type	Description
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
TaskName	string	The name of task, unique identifier of service
TaskAuthorName	string	Username of task author
TaskUserName	string	Username from which the service will be launched
TaskUserId	int	Identifier of user from which the service will be launched. Only for Linux/Mac OS
TaskUserSid	string	SID of user from which the service will be launched. Only for Windows
TaskRunlevel	string	The level of privileges for task running
TaskCreationtime	int	Date and time of task creation
TaskCmdline	string	Actions performed in task
TaskEnabled	string	Flag whether the task is enabled

Field	Type	Description
TaskHidden	string	Flag whether the task is hidden
TaskLocation	string	Path to the task object in the operating system
TaskSchedule	string	Schedule of task running
TaskSource	string	The source of task
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment

Field	Type	Description
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS

Field	Type	Description
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process

Field	Type	Description
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process.
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process.
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process.
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process.
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process.
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified

Field	Type	Description
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File Owner Change

This event type is supported on Windows and Linux operating systems, except for Windows XP x64, Windows Server 2003 x64, and Linux with a kernel version earlier than 4.19.*.

File Owner Change event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)

Field	Type	Description
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileOldOwnerUserName	string	Domain username of previous file owner
FileOldOwnerGroupName	string	Domain group name of previous file owner
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS

Field	Type	Description
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS

Field	Type	Description
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id

Field	Type	Description
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process.
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process.
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified

Field	Type	Description
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS

Field	Type	Description
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process.
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process.
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process.
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

File ACL Change

This event type is supported on Windows and Linux operating systems, except for Windows XP x64, Windows Server 2003 x64, and Linux with a kernel version earlier than 4.19.*.

File ACL Change event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base

Field	Type	Description
AgentIndexTimestamp	int	Timestamp of last bases update for all components
FileDacl	string	The list of access control entries defined in the DACL of the file
FileOldDacl	string	The previous list of access control entries defined in the DACL of the file
FileModifiedDacl	string	The list of modified entries from the DACL of the file
FileSacl	string	The list of access control entries defined in the SACL of the file
FileOldSacl	string	The previous list of access control entries defined in the SACL of the file
FileModifiedSacl	string	The list of modified entries from the SACL of the file
FileName	string	Name of the file (without the path)
FilePath	string	Full path to the file (without file name)
FileFullname	string	Full path to the file
FileMd5	string	MD5 hash of the file content
FileSha256	string	SHA256 hash of the file content
FileSize	int	Size of the file in bytes
FileCreationtime	int	Creation time of the file
FileModificationtime	int	The last time when the file was modified
FileOwnerUserId	int	File owner user id. Specified for Linux and macOS
FileOwnerUserName	string	File owner username. Specified for Linux and macOS
FileOwnerGroupName	string	File owner group name. Specified for Linux and macOS
FileOwnerGroupId	int	File owner group id. Specified for Linux and macOS
FileType	string	Type of the file
FileCapPermitted	string	Permitted capabilities of the file
FileCapInheritable	string	Inherited capabilities of the file
FileCapEffective	string	Effective capabilities of the file
FileOpenflags	string	Specified open flags (for example O_RDWR | O_APPEND), for create/modify/read operations.
FileAttrModificationtime	int	The last time, when the file attributes were modified
FileAttr	int	The file system attributes
FileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier of the file
FileProductname	string	Product name value from the VERSIONINFO resource of the file
FileProductvendor	string	Company name value from the VERSIONINFO resource of the file
FileDescription	string	Description from the VERSIONINFO resource of the file
FileOriginalname	string	Original file name from the VERSIONINFO resource of the file
FileSignSubjectname	string	Information about the organization that signed the file
FileSignResult	string	The result of the signature verification: passed - true, otherwise - false
FileSignTimestamp	int	Date and time when the file was signed
FileUrl	string	URL from where the file was downloaded
FileEmailSrcAddress	string	The sender's address from which the file was received as an attachment
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags

Field	Type	Description
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file

Field	Type	Description
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)

Field	Type	Description
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process

Field	Type	Description
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process.
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process.
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process

Field	Type	Description
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process.
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process.
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process.
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file

Field	Type	Description
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry Owner Change

This event type is supported on Windows operating systems except for Windows XP x64 and Windows Server 2003 x64.

Registry Owner Change event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack

Field	Type	Description
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegOwnerUserName	string	Domain username of key registry owner
RegOldOwnerUserName	string	Domain username of previous key registry owner
RegOwnerGroupName	string	Domain group name of key registry owner
RegOldOwnerGroupName	string	Domain group name of previous key registry owner
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)

Field	Type	Description
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process

Field	Type	Description
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process

Field	Type	Description
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false

Field	Type	Description
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Registry ACL Change

This event type is supported on Windows operating systems except for Windows XP x64 and Windows Server 2003 x64.

Registry ACL Change event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server

Field	Type	Description
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
RegKeyName	string	Full path and name of the registry key which has been affected by the operation
RegDacl	string	The list of access control entries defined in the DACL of the registry key
RegOldDacl	string	The previous list of access control entries defined in the DACL of the registry key
RegModifiedDacl	string	The list of modified entries from the DACL of the registry key
RegSacl	string	The list of access control entries from SACL registry key
RegOldSacl	string	The previous list of access control entries defined in the SACL of the registry key
RegModifiedSacl	string	The list of modified entries from the SACL of the registry key
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)

Field	Type	Description
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id

Field	Type	Description
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process

Field	Type	Description
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file

Field	Type	Description
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Executable Memory Mapped

This event type is supported on Linux operating systems.

Executable Memory Mapped event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC

Field	Type	Description
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
SyscallArgs	string	System call arguments. Specified for Linux
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of LNK file
ProcessFileLnkMd5	string	MD5 hash of the LNK file that was executed
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	File name of executed process (without the path)
ProcessFilePath	string	Full path to the process of executed file (without file name)
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	File size of executed process (in bytes)
ProcessFileCreationtime	int	Creation time of the process executable file

Field	Type	Description
ProcessFileModificationtime	int	The last time, when the executed file of current process was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the file that was executed
ProcessFileSha256	string	SHA256 hash of the process executed file
ProcessFileAttr	int	The list of file system attributes of the executable file from which the process was started
ProcessFileType	string	File type of executed process
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executed file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The name of the entity (person or organization) that issued the certificate
ProcessLogonType	string	Type of logon session of the user who has executed the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	string	The type of account of the user who started the process. Possible values: Unknown, Admin, NotAdmin
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	The fully qualified domain name (FQDN) of the remote host from which the user who started the process logged in
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed

Field	Type	Description
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	string	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process.
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process.
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process.
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded

Field	Type	Description
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	string	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process.
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process.
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process.

Field	Type	Description
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process.
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	string	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Agent Start

This event type is supported on Windows, Linux, and macOS operating systems.

Agent Start event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
InitiatorId	string	Event initiator identifier
InitiatorName	string	Name of the event initiator
InitiatorType	string	Type of the event initiator
AgentInstanceid	string	Identifier of unique agent run
ProcessId	int	Process id
ProcessUniqueid	int	Process unique id

Agent Stop

This event type is supported on Windows, Linux, and macOS operating systems.

Agent Stop event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
InitiatorId	string	Event initiator identifier
InitiatorName	string	Name of the event initiator
InitiatorType	string	Type of the event initiator
AgentInstanceid	string	Identifier of unique agent run
ProcessId	int	Agent process id
ProcessUniqueid	int	Agent process unique id
AgentStopreason	string	Reason for agent termination
AgentExitcode	int	Agent process exit code

Agent Install

This event type is supported on Windows, Linux, and macOS operating systems.

Agent Install event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
InitiatorId	string	Event initiator identifier
InitiatorName	string	Name of the event initiator
InitiatorType	string	Type of the event initiator

Agent Uninstall

This event type is supported on Windows, Linux, and macOS operating systems.

Agent Uninstall event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
InitiatorId	string	Event initiator identifier
InitiatorName	string	Name of the event initiator
InitiatorType	string	Type of the event initiator

Agent Action

This event type is supported on Windows, Linux, and macOS operating systems.

Agent Action event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components

Field	Type	Description
InitiatorId	string	Event initiator identifier
InitiatorName	string	Name of the event initiator
InitiatorType	string	Type of the event initiator
AgentInstanceid	string	Identifier of unique agent run
ProcessId	int	Agent process id
ProcessUniqueid	int	Agent process unique id
AgentActionType	string	The type of action performed in the agent:0 → SelfProtectionDisabled,1 → SelfProtectionEnabled,2 → SelfProtectionTriggered,3 → PasswordProtectionOn,4 → PasswordProtectionOff,5 → CheckedPermission,Otherwise → Unknown
AgentActionDescription	string	Description of the performed action
AgentResultcode	int	The result code of the performed action

System Startup

This event type is supported on Windows, Linux, and macOS operating systems.

System Startup event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated

Field	Type	Description
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components

System Shutdown

This event type is supported on Windows, Linux, and macOS operating systems.

System Shutdown event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux, macOS)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname, where the event was generated
HostIp	string	IP address of the host
AgentVersion	string	The version of product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components

Connection Close

Enabling the `Connection Close` event may result in a significant increase in resource usage on protected devices.

This event type is supported on Windows and Linux operating systems.

Connection Close event fields

Field	Type	Description
IoAId	string array	Indicator of attack unique identifier
IoAName	string array	The name of indicator of attack
IoAVersion	integer array	Indicator of attack version
IoAAlert	string array	The attribute of raising alert
IoAConfidence	string array	The level of indicator of attack detection confidence (High, Medium, Low)
IoASeverity	string array	The level of indicator of attack severity (High, Medium, Low)
IoATactics	string array	Tactics of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoATechniques	string array	Techniques of the MITRE ATT&CK® Matrix on which the current Indicator of attack was mapped
IoAEventIds	string array	The list of event identifiers related to the current Indicator of attack
IoAEventCount	integer array	The count of events related to the current Indicator of attack
IoASB	string array	Full path to file which will be send to Sandbox according to Indicator of attack logic
EventType	string	Notification type
EventId	string	Unique event id
EventTimestamp	int	Date and time on the client (originally in UTC)
EventReceivedtimestamp	int	Date and time when notification was received on server
EventSequenceid	int	The sequence number of the sent event
HostId	string	Host unique id
HostKscId	string	Host id for administration via KSC
HostMdrId	string	Protected host ID in the MDR solution
HostOsmpId	string	List of identifiers for OSMP Based XDR solution servers associated with the protected host
HostOsFamily	string	Operation system family (Windows, Linux)
HostOsName	string	Operation system name
HostOsVersion	string	Operation system version
HostName	string	Hostname generated event
HostIp	string	IP address of the host

Field	Type	Description
AgentVersion	string	Product that is installed on the computer
AgentKdbTimestamp	int	Timestamp of component update KDB
AgentSw2Timestamp	int	Timestamp of component update sw2 base
AgentIndexTimestamp	int	Timestamp of last bases update for all components
NetSrcIp	string	Source ipv4/ipv6 address
NetSrcPort	int	Source port number
NetDstIp	string	Destination ipv4/ipv6 address
NetDstPort	int	Destination port number
NetProtocol	string	IP protocol
NetDirection	string	Direction of network connection (inbound, outbound)
NetCreationtime	int	Creation time of the connection
NetBytessent	int	Number of bytes sent during the connection
NetBytesreceived	int	Number of bytes received during the connection
NetDuration	int	Duration of the connection in seconds
ProcessDynamicstate	int	Terminated - 0x1, Remote threads created - 0x2, Remote memory writes possible - 0x4, Has visible windows - 0x8
ProcessUniqueid	int	Process unique id
ProcessId	int	Process id
ProcessCmdline	string	Command line of the process
ProcessCreationflags	string	List of process creation flags
ProcessDirectory	string	Process initial directory
ProcessEnvironment	string	Environment variables
ProcessProperties	string	List of properties of the account that runs process
ProcessFileLnkName	string	Name of the LNK file that was executed (without the path)
ProcessFileLnkPath	string	Full path to the LNK file that was executed (without file name)
ProcessFileLnkFullname	string	Full path to the LNK file that was executed
ProcessFileLnkCmdline	string	Command line of the executed LNK file
ProcessFileLnkMd5	string	MD5 hash of the executed LNK file
ProcessFileLnkSha256	string	SHA256 hash of the executed LNK file
ProcessIntegritylevel	int	Process integrity level
ProcessType	string	Type of the process
ProcessUserRealid	int	Real user id. Specified for Linux and macOS
ProcessGroupRealid	int	Real user group id. Specified for Linux and macOS
ProcessUserRealname	string	Real Username. Specified for Linux and macOS
ProcessGroupRealname	string	Real group name. Specified for Linux and macOS
ProcessUserEffectiveid	int	Effective user id. Specified for Linux and macOS
ProcessGroupEffectiveid	int	Effective user group id. Specified for Linux and macOS
ProcessUserEffectivename	string	Effective username mapped. Specified for Linux and macOS
ProcessGroupEffectivename	string	Effective group name mapped. Specified for Linux and macOS
ProcessFileName	string	Name of the process executable file (without the path)
ProcessFilePath	string	Full path to the process executable file (without file name)

Field	Type	Description
ProcessFileFullname	string	Full path to the process executable file
ProcessFileSize	int	Process executable file size in bytes
ProcessFileCreationtime	int	Creation time of the process executable file
ProcessFileModificationtime	int	The last time, when the process executable file was modified
ProcessFileAttrModificationtime	int	The last time, when the process executable file attributes were modified
ProcessFileMd5	string	MD5 hash of the process executable file
ProcessFileSha256	string	SHA256 hash of the process executable file
ProcessFileAttr	int	The list of file system attributes for the process executable file
ProcessFileType	string	Type of the process executable file
ProcessFileUrl	string	URL from where the process executable file was downloaded
ProcessFileEmailSrcAddress	string	The sender's address from which the process executable file was received as an attachment
ProcessFileOwnerUserId	int	File owner user ID
ProcessFileOwnerGroupId	int	File owner group ID
ProcessFileOwnerUserName	string	File owner username
ProcessFileOwnerGroupName	string	File owner group name
ProcessFileCapPermitted	string	Permitted capabilities of the process executable file
ProcessFileCapInheritable	string	Inheritable capabilities of the process executable file
ProcessFileCapEffective	string	Effective capabilities of the process executable file
ProcessFileZoneidentifier	int	Zone id given from the alternative stream Zone.Identifier connected with the file
ProcessFileProductname	string	The name of the application from the VERSIONINFO resource
ProcessFileProductversion	string	The application version from the VERSIONINFO resource
ProcessFileOriginalname	string	Original file name from the VERSIONINFO resource
ProcessFileProductvendor	string	Company name value from the VERSIONINFO resource
ProcessFileDescription	string	Description from the VERSIONINFO resource
ProcessFileVersion	string	File version from the VERSIONINFO resource
ProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ProcessFileSignTimestamp	int	Date and time when the file was signed
ProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ProcessLogonType	string	Type of logon session of the process
ProcessLogonSessionId	int	Logon session ID from the access token of the process
ProcessUserType	int	List of properties of the account which runs the process
ProcessUserName	string	Domain and username who executed the process
ProcessLogonUserId	int	Login user id. Specified for Linux and macOS
ProcessLogonHostName	string	Logon remote host/ip. Specified for Linux and macOS
ParentProcessUniqueid	int	Parent process unique id
ParentProcessId	int	Parent process id
ParentProcessCmdline	string	Command line of the parent process
ParentProcessCreationflags	string	Creation flags of parent process
ParentProcessDirectory	string	Current working directory of parent process. Specified for Linux and macOS
ParentProcessFileLnkName	string	Name without the path of the LNK file that started the parent process

Field	Type	Description
ParentProcessFileLnkPath	string	Full path without the name of the LNK file that started the parent process
ParentProcessFileLnkFullname	string	Full path to the LNK file that started the parent process
ParentProcessFileLnkCmdline	string	Command line for LNK file that started the parent process
ParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the parent process
ParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the parent process
ParentProcessIntegritylevel	int	Integrity level of the parent process
ParentProcessType	string	Type of the parent process
ParentProcessUserRealid	int	Real user id of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealid	int	Real user group id of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserRealname	string	Real username of the user who started the parent process. Specified for Linux and macOS
ParentProcessGroupRealname	string	Real group name of the user who started the parent process. Specified for Linux and macOS
ParentProcessUserEffectiveid	int	Effective user ID of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectiveid	int	Effective group ID of the user who started the parent process. Auditd reference field
ParentProcessUserEffectivename	string	Effective username of the user who started the parent process. Auditd reference field
ParentProcessGroupEffectivename	string	Effective group name of the user who started the parent process. Auditd reference field
ParentProcessFileName	string	Name of the parent process executable file (without the path)
ParentProcessFilePath	string	Full path to the parent process executable file (without file name)
ParentProcessFileFullname	string	Full path to the parent process executable file
ParentProcessFileCreationtime	int	Creation time of the parent process executable file
ParentProcessFileModificationtime	int	The last time, when the parent process executable file was modified
ParentProcessFileAttrModificationtime	int	The last time, when the parent process executable file attributes were modified
ParentProcessFileMd5	string	MD5 hash of the parent process executable file
ParentProcessFileSha256	string	SHA256 hash of the parent process executable file
ParentProcessFileAttr	int	The list of file system attributes for the parent process executable file
ParentProcessFileType	string	Type of the parent process executable file
ParentProcessFileUrl	string	URL from where the parent process executable file was downloaded
ParentProcessFileEmailSrcAddress	string	The sender's address from which the parent process executable file was received as an attachment
ParentProcessFileOwnerUserId	int	User ID of the owner of the parent process executable file
ParentProcessFileOwnerGroupId	int	Group ID of the owner of the parent process executable file
ParentProcessFileOwnerUserName	string	Username of the owner of the parent process executable file
ParentProcessFileOwnerGroupName	string	Group name of the owner of the parent process executable file
ParentProcessFileCapEffective	string	Effective capabilities of the parent process executable file
ParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the parent process executable file
ParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the parent process executable file
ParentProcessFileDescription	string	Description from the VERSIONINFO resource of the parent process executable file

Field	Type	Description
ParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
ParentProcessFileSignTimestamp	int	Date and time when the file was signed
ParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
ParentProcessLogonType	string	Type of logon session of the parent process
ParentProcessLogonSessionId	int	Logon session ID from the access token of the parent process
ParentProcessProperties	int	List of properties of the account which runs parent process
ParentProcessUserName	string	Domain and username who executed parent process
GrandParentProcessUniqueid	int	Grand-parent process unique id
GrandParentProcessId	int	Grand-parent process id
GrandParentProcessCmdline	string	Command line of the grand-parent process
GrandParentProcessCreationflags	string	Grand-parent process creation flags
GrandParentProcessDirectory	string	Current working dir of grand-parent process. Linux or Macos specific field
GrandParentProcessFileLnkName	string	Name of the LNK file that started the grand-parent process (without the path)
GrandParentProcessFileLnkPath	string	Full path to the LNK file that started the grand-parent process (without file name)
GrandParentProcessFileLnkFullname	string	Full path to the LNK file that started the grand-parent process
GrandParentProcessFileLnkCmdline	string	Command line for LNK file that started the grand-parent process
GrandParentProcessFileLnkMd5	string	MD5 hash of LNK file that started the grand-parent process
GrandParentProcessFileLnkSha256	string	SHA256 hash of LNK file that started the grand-parent process
GrandParentProcessIntegritylevel	int	Integrity level of the grand-parent process
GrandParentProcessType	string	Type of the grand-parent process
GrandParentProcessUserRealid	int	Real user id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealid	int	Real user group id of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserRealname	string	Real username of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessGroupRealname	string	Real group name of the user who started the grand-parent process. Specified for Linux and macOS
GrandParentProcessUserEffectiveid	int	Effective user ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectiveid	int	Effective group ID of the user who started the grand-parent process. Auditd reference field
GrandParentProcessUserEffectivename	string	Effective username of the user who started the grand-parent process. Auditd reference field
GrandParentProcessGroupEffectivename	string	Effective group name of the user who started the grand-parent process. Auditd reference field
GrandParentProcessFileName	string	Name of the grand-parent process executable file (without the path)
GrandParentProcessFilePath	string	Full path to the grand-parent process executable file (without file name)
GrandParentProcessFileFullname	string	Full path to the grand-parent process executable file
GrandParentProcessFileCreationtime	int	Creation time of the grand-parent process executable file
GrandParentProcessFileModificationtime	int	The last time, when the grand-parent process executable file was modified
GrandParentProcessFileAttrModificationtime	int	The last time, when the grand-parent process executable file attributes were modified
GrandParentProcessFileMd5	string	MD5 hash of the grand-parent process executable file
GrandParentProcessFileSha256	string	SHA256 hash of the grand-parent process executable file
GrandParentProcessFileAttr	int	The list of file system attributes for the grand-parent process executable file

Field	Type	Description
GrandParentProcessFileType	string	Type of the grand-parent process executable file
GrandParentProcessFileUrl	string	URL from where the grand-parent process executable file was downloaded
GrandParentProcessFileEmailSrcAddress	string	The sender's address from which the grand-parent process executable file was received as an attachment
GrandParentProcessFileOwnerUserId	int	User ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupId	int	Group ID of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerUserName	string	Username of the owner of the grand-parent process executable file
GrandParentProcessFileOwnerGroupName	string	Group name of the owner of the grand-parent process executable file
GrandParentProcessFileCapEffective	string	Effective capabilities of the grand-parent process executable file
GrandParentProcessFileProductname	string	Product name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileDescription	string	Description from the VERSIONINFO resource of the grand-parent process executable file
GrandParentProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
GrandParentProcessFileSignTimestamp	int	Date and time when the file was signed
GrandParentProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
GrandParentProcessLogonType	string	Type of logon session of the grand-parent process
GrandParentProcessLogonSessionId	int	Logon session ID from the access token of the grand-parent process
GrandParentProcessProperties	int	List of properties of the account which runs grand-parent process
GrandParentProcessUserName	string	Domain and username who executed grand-parent process
DownloaderProcessUniqueid	int	Downloader process unique id
DownloaderProcessId	int	Downloader process id
DownloaderProcessCmdline	string	Command line of the downloader process
DownloaderProcessCreationflags	string	Downloader process creation flags
DownloaderProcessDirectory	string	Current working dir of downloader process. Linux or Macos specific field
DownloaderProcessFileLnkName	string	Name of the LNK file that started the downloader process (without the path)
DownloaderProcessFileLnkPath	string	Full path to the LNK file that started the downloader process (without file name)
DownloaderProcessFileLnkFullname	string	Full path to the LNK file that started the downloader process
DownloaderProcessFileLnkCmdline	string	Command line for LNK file that started the downloader process
DownloaderProcessFileLnkMd5	string	MD5 hash of LNK file that started the downloader process
DownloaderProcessFileLnkSha256	string	SHA256 hash of LNK file that started the downloader process
DownloaderProcessIntegritylevel	int	Integrity level of the downloader process
DownloaderProcessType	string	Type of the downloader process
DownloaderProcessUserRealid	int	Real user id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealid	int	Real user group id of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessUserRealname	string	Real username of the user who started the downloader process. Specified for Linux and macOS
DownloaderProcessGroupRealname	string	Real group name of the user who started the downloader process. Specified for Linux and macOS

Field	Type	Description
DownloaderProcessUserEffectiveid	int	Effective user ID of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectiveid	int	Effective group ID of the user who started the downloader process. Auditd reference field
DownloaderProcessUserEffectivename	string	Effective username of the user who started the downloader process. Auditd reference field
DownloaderProcessGroupEffectivename	string	Effective group name of the user who started the downloader process. Auditd reference field
DownloaderProcessFileName	string	Name of the downloader process executable file (without the path)
DownloaderProcessFilePath	string	Full path to the downloader process executable file (without file name)
DownloaderProcessFileFullname	string	Full path to the downloader process executable file
DownloaderProcessFileCreationtime	int	Creation time of the downloader process executable file
DownloaderProcessFileModificationtime	int	The last time, when the downloader process executable file was modified
DownloaderProcessFileAttrModificationtime	int	The last time, when the downloader process executable file attributes were modified
DownloaderProcessFileMd5	string	MD5 hash of the downloader process executable file
DownloaderProcessFileSha256	string	SHA256 hash of the downloader process executable file
DownloaderProcessFileAttr	int	The list of file system attributes for the downloader process executable file
DownloaderProcessFileType	string	Type of the downloader process executable file
DownloaderProcessFileUrl	string	URL from where the downloader process executable file was downloaded
DownloaderProcessFileEmailSrcAddress	string	The sender's address from which the downloader process executable file was received as an attachment
DownloaderProcessFileOwnerUserId	int	User ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupId	int	Group ID of the owner of the downloader process executable file
DownloaderProcessFileOwnerUserName	string	Username of the owner of the downloader process executable file
DownloaderProcessFileOwnerGroupName	string	Group name of the owner of the downloader process executable file
DownloaderProcessFileCapEffective	string	Effective capabilities of the downloader process executable file
DownloaderProcessFileProductname	string	Product name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileDescription	string	Description from the VERSIONINFO resource of the downloader process executable file
DownloaderProcessFileSignResult	string	The result of the signature verification: passed - true, otherwise - false
DownloaderProcessFileSignTimestamp	int	Date and time when the file was signed
DownloaderProcessFileSignSubjectname	string	The subject name of the certificate which was used to sing the file
DownloaderProcessLogonType	string	Type of logon session of the downloader process
DownloaderProcessLogonSessionId	int	Logon session ID from the access token of the downloader process
DownloaderProcessProperties	int	List of properties of the account which runs downloader process
DownloaderProcessUserName	string	Domain and username who executed downloader process
CreatorProcessUniqueid	int	Unique ID of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessId	int	Id of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessCmdline	string	Command line of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessCreationflags	string	Creation flags of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessDirectory	string	Current working directory of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkName	string	Name of the LNK file (without the path) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkPath	string	Full path to the LNK file (without the name) that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkFullname	string	Full path to the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkCmdline	string	Command line parameters of the LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkMd5	string	MD5 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileLnkSha256	string	SHA256 hash of LNK file that started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessIntegritylevel	int	Integrity level of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessType	string	Type of the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealid	int	Real user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealid	int	Real group ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserRealname	string	Real username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupRealname	string	Real group name of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectiveid	int	Effective user ID of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessGroupEffectiveid	int	Effective group ID of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserEffectivename	string	Effective username of the user who started the process that was real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessGroupEffectivename	string	Effective group name of the user who started the process that was real initiator of the current process was downloaded. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileName	string	Name of the executable file of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFilePath	string	Full path (without file name) to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileFullname	string	Full path to the executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCreationtime	int	Creation time of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileModificationtime	int	Modification time of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttrModificationtime	int	The modification time of the attributes of the executable file whose process is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileMd5	string	MD5 hash of executable file of process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSha256	string	SHA256 hash of executable file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileAttr	int	The list of file system attributes for the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileType	string	Type of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileUrl	string	URL from which the process that is real initiator of the current process was downloaded. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileEmailSrcAddress	string	The sender's address from which the executable file was received as an attachment, which is the actual initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserId	int	Owner's User ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Field	Type	Description
CreatorProcessFileOwnerGroupId	int	Owner's Group ID of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerUserName	string	Owner's Username of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOwnerGroupName	string	Owner's Group name of the process file that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileCapEffective	string	Effective capabilities of the process that is real initiator of the current process. Specified for Linux and macOS. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductname	string	Product name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileOriginalname	string	Original file name from the VERSIONINFO resource of the process file that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileProductvendor	string	Company name value from the VERSIONINFO resource of the process file that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileDescription	string	Description from the VERSIONINFO resource of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignResult	string	The result of the signature verification for executable file of the process that is real initiator of the current process: passed - true, otherwise - false. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignTimestamp	int	Date and time when the file (real initiator of the current process) was signed. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessFileSignSubjectname	string	The subject name of the certificate which was used to sign the file of the process that is real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonType	string	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessLogonSessionId	int	Logon session ID from the access token of the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessProperties	int	Logon session type for the user who started the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'. Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields
CreatorProcessUserName	string	Domain and username of the user who strated the process that was real initiator of the current process. Typically, 'ParentProcess' is the same as 'CreatorProcess'.Then the CreatorProcess* fields will be absent in the telemetry and it is necessary to look at the ParentProcess* fields

Granular access to events

In KUMA, users with different rights can have granular access to events. Access to events is controlled at the level of storage spaces.

You can assign spaces to users in the **Spaces permissions** section. After upgrading to the latest version, the 'All spaces' space set is assigned to all existing users, that is, access to all spaces is unrestricted. An event contains a tenant ID and a space ID, therefore the user needs rights to the corresponding tenant and space to have access to the event.

Keep in mind the following special considerations involved in displaying storages:

- If a storage is not listed in the **Active services** section, the storage and its spaces are not displayed in the list of spaces of the set.
- If the storage service was stopped using the `systemctl stop kuma-<storage ID>` command, the storage and its spaces are not displayed in the list of spaces of the set.
- If the storage was started and then deleted using the `uninstall` command, the storage and its spaces remain in the list of spaces of the set.

In the list of events, you can add the **SpaceID** field to the table, which will display the name of the space. The space of audit events is displayed as KUMA Audit. KUMA Default is the space inside each storage, where all events go if the storage does not have configured spaces or if the event does not match the conditions of the existing spaces.

When you export the list of events to a TSV file, the space ID and name are displayed for spaces.

To differentiate access:

1. Configure the space sets.

   You can create, edit, or delete space sets. These actions result in audit events.

2. Configure the access rights of the space set: you can grant or revoke access rights of selected users.

Creating a space set

To create a space set:

1. In the KUMA Console, go to the **Settings** → **Spaces permissions** section.
2. In the **Spaces permissions** window, click the **Add** button.
3. In the **Create space set** window, configure the following:
   a. In the **Name** field, specify the space set name. You can use up to 128 Unicode characters.
   b. In the **Spaces** drop-down list, select the spaces that you want to include in the set by selecting the check box next to the space name. To remove a space from the set, clear the check box.
   c. If you want the default space set to be available to all users that do not have any of the sets assigned, select the **Make default** check box.

This makes the created set the default set. In the list of space sets, it is marked as the default set: in the **Default** column, it has **Yes**.

There can be only one default set. If you select **Make default** check box in the settings of a different set and save the settings, the default set is reassigned.

4. Click **Create**.

The space set is created and displayed in the list of spaces. After the space set is created, an audit event is generated in KUMA.

Then you need to grant users access rights to the created space set.

Editing a space set

To edit a space set:

1. In the KUMA Console, go to the **Settings** → **Spaces permissions** section.
2. In the **Spaces** permissions window, click a set of spaces.
3. In the **Edit** space set window, configure the following:
   a. In the **Name** field, specify the space set name. You can use up to 128 Unicode characters.
   b. In the **Spaces** drop-down list, select the spaces that you want to include in the set by selecting the check box next to the space name. To remove a space from the set, clear the check box.
   c. If you want the default space set to be available to all users that do not have any of the sets assigned, select the **Make default** check box.

      This makes the set the default set when you save the changes. In the list of space sets, it is marked as the default set: in the **Default** column, it has **Yes**.

      There can be only one default set. If you select **Make default** check box in the settings of a different set and save the settings, the default set is reassigned.
4. Click **Save**.

The space set is saved and displayed in the list of spaces.

The edited set of spaces is accessible to all users who had access to the original space set. You can control access to the edited space set and grant or revoke access permissions using the **Access control** button.

Deleting a space set

To delete a space set:

1. In the KUMA Console, go to the **Settings** → **Spaces permissions** section.
2. In the **Spaces permissions** window, select the check box next to the relevant space set.

   If you want to delete all space sets, select the **Select all** check box in the upper left part of the list and click **Delete**. The default **All spaces** set cannot be deleted.
3. Click **Delete**.

The space set is deleted and removed from the list of spaces. After the space set is deleted, an audit event is generated in KUMA.

If a user had access only to the deleted space set and did not have access to any other sets, such a user automatically gets access rights to the default set.

Grant access to a space set

To grant access to a space set

1. In the KUMA Console, go to the **Settings** → **Spaces permissions** section.
2. Select a set of spaces in the list by selecting the check box next to it, then click the **Access control** button on the toolbar.
3. This opens the **Users** window; in that window, click **Add users**.
4. In the list in the **Add users** window, select users to which you want to grant access to the space set and click **Add**.

   If you want to grant access to the selected space set to all users, select the check box in the upper left corner and select one of the following options: **Select all in page** or **Select all**, then click **Add**.

   This opens a confirmation window.
5. If you want to grant access to the listed users, click **Grant** in the confirmation window.

The users now have access to the space set.

Revoke access to a space set

To revoke access to a space set:

1. In the KUMA Console, go to the **Settings** → **Spaces permissions** section.
2. Select a set of spaces in the list by selecting the check box next to it, then click the **Access control** button on the toolbar.
3. This opens the **Users** window; in that window, select users by selecting their check boxes, then click **Delete**.

   If you want to revoke access to the selected space set from all users, select the check box in the upper left corner and select one of the following options: **Select all in page** or **Select all**, then click **Delete**.

This opens a confirmation window.

4. If you want to revoke access from the listed users, click **Revoke** in the confirmation window.

   The users no longer have access to the space set.

   If a user does not have access to any other space set, such a user automatically gets access to the default set.

Use cases

Migrating to the latest KUMA version with differentiated access to events

Goal: Update the product.

Action: The administrator updates KUMA to version 3.4.

Result: The accessibility of events does not change for users; all users have access to events in those tenants in which users are allowed to view the list of events. When adding new users, tenants, or spaces, all users see the events of tenants depending on whether they are allowed to view the list of events, with no further restrictions.

Restricting access to spaces for all users

Goal: Some spaces in the installation must be inaccessible to some users. Current and new users must not have access to such spaces.

Action: The administrator creates a new set of spaces that includes the spaces to which all users must have access. Each space is specified as a tuple: clusterID, tenantID, spaceID. In this way, you can assemble a set of spaces from a finite number of spaces. After creating the set, the administrator makes this set the default set.

Result: For all users that do not have this space set explicitly specified, including new users, access to events is additionally restricted in accordance with the default space set. That is, such users have access to events in tenants in which the user has the right to view the list of events, and events in spaces that are included in the default set.

Allowing some users to view all events

Goal: KUMA has a default set in which only a finite number of spaces is explicitly specified. Also some users in KUMA must have unrestricted access to all events. You want to allow these users to view events from spaces that are not specified in the default set.

Action: The administrator goes to the **Spaces permissions** section and selects the **All spaces** set, which gives access to all spaces without restrictions, and grants access to this set to the selected users.

Result: Users that have been explicitly granted access to the **All spaces** set now have unrestricted access to events in spaces. Their access to events depends only on the right to view the list of events in accordance with their assigned role. When you add new spaces or tenants, these users will have access to the events located in these new spaces and tenants, provided they have the right to view the list of events for the corresponding tenant. Users that have not been explicitly granted access to the **All spaces** set will have restricted access to events in accordance with the default set of spaces.

Permitting some users to view events from a finite set of spaces

Goal: KUMA has a default set in which only a finite number of spaces is explicitly specified. Some users need to see a different space set, but not all available spaces.

Action: The administrator creates a space set that includes the relevant spaces and grants access to this set to the selected users.

Result: User access to events is restricted in accordance with the available space set. Users that have not been explicitly granted access to the set will have restricted access to events in accordance with the default set of spaces.

Supplementing an explicitly specified space set for a user

Goal: Users u1, u2, u3, u4 have been explicitly granted access to space set 'set1', which includes spaces to which Windows and Linux events go. Users u1 and u2, in addition to Windows and Linux events, need to view Cisco and VMware events that go to the respective spaces.

Action: The administrator creates a set of spaces, 'set2', which includes the spaces of Cisco and VMware events. The administrator assigns set2 to u1 and u2 in addition to set1.

Result: Users u1 and u2 now have access to events from the spaces of all of the sets available to them, that is, events from the set1 and set2: Windows, Linux, Cisco, VMware. Access of users u3 and u4 to events remains unchanged.

Editing a space set

Goal: Some users have access to a space set (access is granted explicitly or the set is the default set). For all of these users, you need to add access to a certain space that is not in the set, or revoke access to a space that is in the set.

Action: The administrator edits the existing set of spaces by adding or removing the relevant spaces.

Result: All users that have access to the set can now view events in accordance with the updated space set.

Deleting a space set

Goal: The space set is no longer relevant, and it needs to be deleted from KUMA.

Action: The administrator deletes an existing space set. The **All spaces** set cannot be deleted. The default space set cannot be deleted.

Result: The deleted space set is gone from KUMA. All users that have been explicitly granted access to this set of spaces now have access to events in accordance with the space sets that remain available to them. If a user does not have any space sets left to which access has been explicitly granted, such a user's access to events is controlled by the default set.

Viewing the events table

The events table provides you with an overview of all events received by KUMA Core from the data sources. The table displays the list of events filtered according to the executed SQL query.

To view the events table:

1. In the main menu, go to **Monitoring** → **Threat hunting**.

2. If necessary, apply the tenant filter. By default, the tenant filter is disabled and the events table displays the events related to all of the tenants to which you have the **Read** access right. To apply the tenant filter:

   a. Click the link next to the **Tenant filter** setting.

      The tenant filter opens.

   b. Select the check boxes next to the required tenants.

      The events table displays only the events related to the selected tenants.

The events table is displayed. For details about the table columns, relate to the normalized event data model.

Searching and filtering events

To search and filter events, modify an SQL query in the search field, and then click the **Run Query** button. You can enter the SQL query manually or generate it by using the query builder.

Data aggregation and grouping is supported in SQL queries.

You can add filter conditions to an already generated SQL query in the window for viewing statistics, the events table, and the event details area.

To change the filtering settings in the Statistics window

1. Follow the steps to open the events table.

2. Open **Statistics** details area by using one of the following methods:

   - Click the ••• button in the top right corner of the events table, and then select **Statistics**.

   - In the events table, click any value, and then select **Statistics** in the context menu that opens.

The **Statistics** details area appears in the right part of the web interface window.

3. Open the drop-down list of the relevant parameter and hover your mouse cursor over the necessary value.

4. Change the filter settings by doing one of the following:

   - To include only events with the selected value, click the + button.

   - To exclude all events with the selected value, click the — button.

To change the filtering settings in the events table

1. Follow the steps to open the events table.

2. Click an event parameter value in the events table.

3. In the opened menu, select one of the following options:

   - To show only events with the selected value, select **Filter by this value**.

   - To exclude all events with the selected value from the table, select **Exclude from filter**.

To change the filter settings in the event details area

1. Follow the steps to open the events table.

2. Click a relevant event to invoke the event details panel.

3. Change the filter settings by doing one of the following:

   - To include only events with the selected value, click the + button.

   - To exclude all events with the selected value, click the — button.

As a result, the filter settings and the events table are updated, and the new search query is displayed in the upper part of the screen.

> When you switch to the query builder, the parameters of a query entered manually in the search field are not transferred to the builder, so you will need to create your query again. The query created in the builder does not overwrite the query that was entered into the search string until you click the **Apply** button in the builder window.

Click the button to save the current filter.

Manually creating SQL queries

You can use the search string to manually create SQL queries of any complexity to filter events.

Executing an SQL query affects the displayed table columns.

If the SQL query contains the * value, columns specified in the query are added to the table if they were absent. Removing a displayed column from the subsequent queries does not hide the corresponding column.

If the SQL query does not contain the * value, the table only displays columns for the specified fields that conform the normalized event data model. Columns are displayed even if there is no data for them.

To manually generate an SQL query:

1. Follow the steps to open the events table.
2. Enter your SQL query into the input field.
3. Click the **Apply query** button.

   The table displays events that satisfy the criteria of your query. If necessary, you can filter events by period.

To display non-printable characters in the SQL query field, press either of the following key combinations:

- **`Ctrl`**+**`*/Command`**+**`*`**
- **`Ctrl`**+**`Shift`**+**`8/Command`**+**`Shift`**+**`8`**

If you enable the display of non-printable characters in the XDR component, other components (such as KUMA) do not automatically display non-printable characters until you reload the components' browser tabs.

Supported functions and operators

SELECT

Event fields that should be returned.

For **SELECT** fields, the program supports the following functions and operators:

Aggregation functions: **count**, **avg**, **max**, **min**, **sum**.

Arithmetic operators: **+**, **-**, *****, **/**, **<**, **>**, **=**, **!=**, **>=**, **<=**.

You can combine these functions and operators.

If you are using aggregation functions in a query, you cannot customize the events table display, sort events in ascending or descending order, or receive statistics.

FROM

Data source.

WHERE

Conditions for filtering events.

- **AND**, **OR**, **NOT**, **=**, **!=**, **>**, **>=**, **<**, **<=**
- **IN**
- **BETWEEN**

- **LIKE**
- **ILIKE**
- **inSubnet**
- **match** (the re2 syntax of regular expressions is used in queries; special characters must be shielded with "\")

GROUP BY

Event fields or aliases to be used for grouping the returned data.

If you are using data grouping in a query, you cannot customize the events table display, sort events in ascending or descending order, receive statistics, or perform a retrospective scan.

ORDER BY

Columns used as the basis for sorting the returned data.

Possible values:

- **DESC** — descending order.
- **ASC** — ascending order.

OFFSET

Skip the indicated number of lines before printing the query results output.

LIMIT

Number of strings displayed in the table.

The default value is 250.

> When switching to the query builder, the query parameters that were manually entered into the search string are not transferred to the builder, so you will need to create your query again. Also, the query created in the builder does not overwrite the query that was entered into the search string until you click the Apply button in the builder window.

> Aliases must not contain spaces.

Example queries

- **SELECT * FROM `events` WHERE Type IN ('Base', 'Audit') ORDER BY Timestamp DESC LIMIT 250**

  In the events table, all events with the **Base** and **Audit** type are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- **SELECT * FROM `events` WHERE BytesIn BETWEEN 1000 AND 2000 ORDER BY Timestamp ASC LIMIT 250**

  All events of the events table for which the **BytesIn** field contains a value of received traffic in the range from 1,000 to 2,000 bytes are sorted by the **Timestamp** column in ascending order. The number of strings that can be displayed in the table is 250.

- **SELECT * FROM `events` WHERE Message LIKE '%ssh:%' ORDER BY Timestamp DESC LIMIT 250**

  In the events table, all events whose **Message** field contains data corresponding to the defined %ssh:% template in lowercase are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- **SELECT * FROM `events` WHERE inSubnet(DeviceAddress, '00.0.0.0/00') ORDER BY Timestamp DESC LIMIT 250**

  In the events table, all events for the hosts that are in the 00.0.0.0/00 subnet are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- **SELECT * FROM `events` WHERE match(Message, 'ssh.*') ORDER BY Timestamp DESC LIMIT 250**

  In the events table, all events whose **Message** field contains text corresponding to the ssh.* template are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

- **SELECT max(BytesOut) / 1024 FROM `events`**

  Maximum amount of outbound traffic (KB) for the selected time period.

- **SELECT count(ID) AS "Count", SourcePort AS "Port" FROM `events` GROUP BY SourcePort ORDER BY Port ASC LIMIT 250**

  Number of events and port number. Events are grouped by port number and sorted by the **Port** column in ascending order. The number of strings that can be displayed in the table is 250.

  The **ID** column in the events table is named Count, and the **SourcePort** column is named Port.

- **SELECT * FROM `events` WHERE match(Message, 'ssh:\'connection.*') ORDER BY Timestamp DESC LIMIT 250**

  If you want to use a special character in a query, you need to escape this character by placing a backslash (\) character in front of it.

  In the events table, all events whose **Message** field contains text corresponding to the ssh: 'connection' template are sorted by the **Timestamp** column in descending order. The number of strings that can be displayed in the table is 250.

Generating an SQL query using a builder

You can use the query builder to generate an SQL query for filtering events.

Executing an SQL query affects the displayed table columns.

If the SQL query contains the * value, columns specified in the query are added to the table if they were absent. Removing a displayed column from the subsequent queries does not hide the corresponding column.

If the SQL query does not contain the * value, the table only displays columns for the specified fields that conform the normalized event data model. Columns are displayed even if there is no data for them.

To generate an SQL query using the builder:

1. Follow the steps to open the events table.

2. Click the button to open the query builder.

   Generate a search query by providing data in the following parameter blocks:

   - **SELECT**

     Event fields that should be returned. The * value is selected by default, which means that all available event fields must be returned. To adjust the displayed fields, select the desired fields in the drop-down list. Note that Select * increases the duration of the request execution, but eliminates the need to specify the fields in the request.

     When selecting an event field, you can use the field on the right of the drop-down list to specify an alias for the column of displayed data, and you can use the right-most drop-down list to select the operation to perform on the data: **count**, **max**, **min**, **avg**, **sum**.

   - **FROM**

     Data source. Select the **events** value.

   - **WHERE**

     Conditions for filtering events.

     To add conditions and groups, click the **Add condition** and **Add group** buttons. The **AND** operator value is selected by default in a group of conditions. Click the operator value to change it. Available values: **AND**, **OR**, **NOT**.

     To change the structure of conditions and condition groups, use the icon to drag and drop expressions.

     To add filter conditions:

     a. In the drop-down list on the left, select the event field that you want to use for filtering.

     b. Select the necessary operator from the middle drop-down list. The available operators depend on the type of value of the selected event field.

     c. Enter the value of the condition. Depending on the selected type of field, you may have to manually enter the value, select it from the drop-down list, or select it on the calendar.

     To delete filter conditions, click the **X** button. To delete group conditions, click the **Delete group** button.

   - **GROUP BY**

     Event fields or aliases to be used for grouping the returned data.

     If you are using data grouping in a query, you cannot customize the events table display, sort events in ascending or descending order, receive statistics, or perform a retrospective scan.

   - **ORDER BY**

     Columns used as the basis for sorting the returned data. In the drop-down list on the right, you can select the necessary order: **DESC** — descending, **ASC** — ascending.

   - **LIMIT**

     Number of strings displayed in the table.

     The default value is 250.

If you are filtering events by a user-defined period and the number of strings in the search results exceeds the defined value, you can click the **Show next records** button to display additional strings in the table. This button is not displayed when filtering events by the standard period.

3. Click the **Apply** button.

   The current SQL query will be overwritten. The generated SQL query is displayed in the search field.

   To reset the builder settings, click the **Default query** button.

   To close the builder without overwriting the existing query, click the button.

4. Click the **Run query** button to display the data in the table.

   You can also run the query in the background. To do this, click the arrow icon next to the **Run query** button and click the **Run in the background** button. In this case, the following happens:

   - The query is queued for execution in the background, and a task is created in the Detection and response section to run this query.
   - Background queries run in parallel at low priority.
   - A query containing more than 10,000 events cannot be queued for execution in the background.

   The table will display the search results based on the generated SQL query. You can view the result of running the query in the background in the **Detection and response** section.

   If necessary, you can cancel the background execution of a query in the background by clicking the arrow icon next to the **Run query** button and selecting **Cancel query**. When a query is canceled, an audit event is automatically generated.

When switching to another section of the web interface, the query generated in the builder is not preserved. If you return to the Events section from another section, the builder will display the default query.

Viewing event details

To open the event details panel, select an event in the events table in the **Threat hunting** section or in an alert details page.

The **Event details** panel appears in the right part of the web interface window and contains a list of the event parameters with values. In this area you can:

- Include the selected field in the search or exclude it from the search by clicking + or − next to a parameter's value.
- Find similar events and add or delete a prevention rule by clicking the **FileHash** and **DeviceCustomString** values.
- When integrated with Kaspersky CyberTrace and Kaspersky Threat Intelligence Portal, you can add to Internal TI of CyberTrace and show info from Threat Lookup by clicking the **FileHash** and **DeviceCustomString** values.
- View the settings of the service that registered the event by clicking the **Service** value.

- View the list of alerts or incidents related to the device by clicking the **DeviceAssetID** value and selecting **Asset details**. In the **Device details** window that opens, click the **Show related** button and select either **Alerts** or **Incidents**, depending on what you want to view.

- View the list of IOA rules that were triggered when the event was created; view the detailed information about those IOA rules by clicking their names. You can also view the detailed information about an IOA rule in Kaspersky TIP. To do this, in the IOA rule details window, click the **Find in Kaspersky TIP** button.

In the **Event details** panel, the name of the described object is shown instead of its ID in the values of the following settings. If you change the filter settings from the **Event details** panel, the object's ID, and not its name, is added to the SQL query:

- TenantID
- SeriviceID
- DeviceAssetID
- SourceAssetID
- DestinationAssetID
- SourceAccountID
- DestinationAccountID

Saving and selecting events filter configuration

You can save the current filter configuration, including the time-based filter, query builder, and the events table settings, for future use. Saved filter configurations are available to you and other users that have corresponding access rights.

To save the current settings of the filter, query, and period

1. Follow the steps to open the events table.
2. Click the icon next to the search query and select **Save current filter**.
3. In the **New filter** window that opens, enter the name of the filter configuration in the **Name** field. The name must contain 128 Unicode characters or less.
4. In the **Tenant** drop-down list, select the tenant for which to save the created filter.
5. Click **Save**.

   The filter configuration is now saved.

To select a previously saved filter configuration

1. Follow the steps to open the events table.
2. Click the icon next to the search query and select the desired filter.

To save the current settings of the filter, query, and the events table settings

1. Follow the steps to open the events table.
2. Click the gear icon in the panel above the events table.
3. Click **Save current preset**.
4. In the **New preset** window that opens, enter the name of the preset in the **Name** field. The name must contain 128 Unicode characters or less.
5. In the **Tenant** drop-down list, select the tenant for which to save the created preset.
6. Click **Save**.

   The preset configuration is now saved.

To select a previously saved preset

1. Follow the steps to open the events table.
2. Click the gear icon in the panel above the events table. Select the **Presets** tab.
3. Select the desired preset.

To delete a previously saved filter configuration for all users

1. Follow the steps to open the events table.
2. Click the icon next to the search query.
3. Click the icon next to the configuration that you need to delete.
4. Click **OK**.

Filtering events by time range

You can specify the period to display events from.

To filter events by time range:

1. Follow the steps to open the events table.
2. Open the second drop-down list in the upper part of the window.
3. Specify the time range. You can select predefined ranges relative to the current date and time or specify a custom range by using the **Range start** and **Range end** fields or by selecting dates in the calendar.
4. Click the **Apply** button.

Exporting events

You can export information about events to a TSV file. The selection of events that will be exported to a TSV file depends on filter settings. The information is exported from the columns that are displayed in the events table. The columns in the exported file are populated with the available data even if they did not display in the events table in the **Threat hunting** section due to the special features of the SQL query.

To export information about events:

1. Follow the steps to open the events table.
2. Click the ••• button in the top right corner of the events table and select **Export TSV**.

   The new export TSV file task is created in the KUMA **Task Manager** section.
3. In the KUMA section, find the task you created in the **Task Manager** section.
4. Click the task type name and select **Upload** from the drop-down list.

   The TSV file will be downloaded using your browser's settings. By default, the file name is event-export-<date>_<time>.tsv.

The file is saved based on your web browser's settings.

Retrospective scan

You can use retrospective scan to refine the correlation rule resources or analyze historical data.

You can also choose to create alerts based on a retrospective scan.

To use retrospective scan:

1. In the main menu, go to **Monitoring** → **Threat hunting**.
2. Click the ••• button in the top right corner of the events table, and then select **Retroscan**.

   The **Retroscan** panel opens.
3. In the **Correlator** drop-down list, select the Correlator to feed selected events to.
4. In the **Correlation rules** drop-down list, select the Correlation rules that must be used when processing events.
5. To execute responses during event processing, turn on the **Execute responses** toggle switch.
6. To generate alerts during event processing, turn on the **Create alerts** toggle switch.
7. Click the **Create task** button.

   The retrospective scan task is created in the KUMA **Task Manager** section.

Getting events table statistics

You can get statistics for the current events selection displayed in the events table. The selected events depend on the filter settings.

To obtain statistics:

1. Follow the steps to open the events table.

2. Do one of the following:

   - In the upper-right corner of the events table, select **Statistics** from the ••• drop-down list.

   - In the events table, click on any value and select **Statistics** from the opened context menu.

The **Statistics** details area appears with the list of parameters from the current event selection. The numbers near each parameter indicate the number of events with that parameter in the selection. If a parameter is expanded, five most frequently occurring values are displayed. Type a parameter name in **Search fields** to filter displayed data.

The **Statistics** window allows you to modify the events filter.

> When using SQL queries with data grouping and aggregation for filtering events, statistics are not available.

Threat response

To perform response actions, view the result of an enrichment that you performed from the playbook, and launch playbooks manually, you have to go to the **Alerts** or **Incidents** sections.

The **Alerts** and **Incidents** sections are displayed in the main menu if the following conditions are met:

- You have a license key for Kaspersky Next XDR Expert.
- You are connected to the root Administration Server in OSMP Console.
- You have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Auditor.

After you perform a response action, you can view the response history.

Response actions

The response actions can be launched in one of the following ways:

- Manually, as described in this section.

  For the devices on which Kaspersky Endpoint Security with a built-in agent is not installed, you cannot perform the following response actions: Managing services, Deleting a file, YARA scan, IOC scan, Getting file from quarantine, and the response actions described in the Getting files for further analysis section.

  When responding from the alert or incident details or from the table of assets, you cannot select archived devices.

- Within a playbook.

  In this case, when creating or editing a playbook you can configure the response action to run automatically, or to request the user's manual approval before launching within the playbook. By default, manual approval of response actions is disabled.

  If a device becomes archived while performing a response action, the playbook result displays the **Error** message. If a playbook is triggered for an alert that includes archived devices, the playbook result displays the **Warning** message and the description that the device is archived.

Terminating processes

The *Terminate process* response action allows you to remotely terminate processes on devices. You can run the Terminate process response action for observables or assets.

You can run the Terminate process response action in one of the following ways:

- From alert or incident details
- From a device details

- From a telemetry event

  You can open the event details from alert or incident details or from the Threat hunting section.

- From an investigation graph

  > The option is available if the investigation graph is built.

- From the table of assets

- From the **Response history** section

You can also configure the response action to run automatically when creating or editing a playbook.

> To run the Terminate process response action, you must have one of the following XDR roles: Main administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

> It might take up to 15 minutes to launch a response action due to the synchronization interval between the managed device and Administration Server.

Responding from the alert or incident details

To terminate process from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the link with the alert ID you need.

   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.

2. In the window that opens, go to the **Assets** tab.

   > You can also go to the **Observables** tab, and select one or several observables for which you want to terminate the process. The observables may include MD5 or SHA256 hashes.

3. In the list of assets, select one or several devices.

4. In the **Select response actions** drop-down list, select **Terminate process**.

5. In the right pane that opens, do the following:

   a. In the **Full path** field, specify the full path to the process file.

   b. In the **Assets** drop-down list, select the devices for which you want to terminate the process.

      If at step 2 you selected the **Assets** tab, the names of the devices whose check boxes you selected are specified in the **Assets** field by default. You can either change them or select the **All assets** option in the **Scope** drop-down list to terminate process on all the devices that belong to the alert or incident tenant and its child tenants.

Only the assets on which you can perform the response action are available for selection.
The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

c. If necessary, configure the following settings in the corresponding fields:

- Specify either the SHA256 or MD5 hash of the process file.

  If at step 2 you selected the **Observables** tab, the hashes that you selected are specified by default. You can change them.

- Write a description or a comment to the response action.

6. Click the **Terminate** button.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the device details

To terminate process from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device on which the file to be moved to quarantine is located.

   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which the file to be moved to quarantine is located.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. Perform the same actions as described at steps 4–6 in *Responding from the alert or incident details* procedure.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the telemetry event

To terminate process from the telemetry event when opening it from the alert or incident details:

1. In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device on which the file to be moved to quarantine is located.

2. In the window that opens, go to the **Details** tab, and do one of the following:

   - Click the name of the required event and select the device.

   - Click the **Find in Threat Hunting** button to go to the **Threat hunting** section and select the required device.

   > You can also go to the **Observables** tab, and select one or several observables for which you want to terminate the process. The observables may include MD5 or SHA256 hashes.

3. In the **Select response actions** drop-down list, select **Terminate process**.

4. Perform the same actions as described at steps 4–6 in the *Responding from the alert or incident details* procedure.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

To respond from a telemetry event when opening it from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting.**

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select **Terminate process**.

4. In the right pane that opens, configure the response action settings as described at steps 5–6 in the *Responding from the alert or incident details* procedure.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the investigation graph

> The option is available if the investigation graph is built.

To terminate process from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.

2. In the **Incident details** window that opens, click the **View on graph** button.

   The **Investigation graph** window opens.

3. Click the name of the required device, and then in the drop-down list, select **View details**.

4. Perform the same actions as described at steps 4–6 in *Responding from the alert or incident details* procedure.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the table of assets

To perform the response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   The archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select **Terminate process**.

4. In the right pane that opens, configure the response action settings as described at steps 5–6 in the *Responding from the alert or incident details* procedure.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the Response history section

To perform the response action from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select **Terminate process**.

3. In the right pane that opens, configure the response action settings as described at steps 5–6 in the *Responding from the alert or incident details* procedure.

The process is terminated. If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response actions in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

After you click the link in the **Action status** column, the window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, the **Error** message and the information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Moving devices to another administration group

As a response action, you can move a device to another administration group of Open Single Management Platform. This may be required when the analysis of an alert or incident shows that the protection level of the device is low. When you move a device to another administration group, the group policies and tasks are applied to the device.

> The administration group to which you move the device must belong to the same tenant as the device.

You can move a device to another administration group in one of the following ways:

- From the alert or incident details
- From the device details
- From an investigation graph

You can also configure the response action to run automatically when creating or editing a playbook.

> To move a device to another administration group, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

> It might take up to 15 minutes to launch a response action due to the synchronization interval between the managed device and Administration Server.

Moving a device to another administration group from alert or incident details

To move a device to another administration group from alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device to be moved.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device to be moved.
2. In the window that opens, go to the **Assets** tab.

3. Select check box next to the device to be moved to another administration group.

> You can select several devices, if the devices are managed by the same Administration Server: primary, secondary, or virtual.

4. In the **Select response actions** drop-down list, select **Move to group**.

   The **Move to group** window that opens on the right side of the screen displays the administration groups of the Administration Server that manages the selected device.

5. Select the administration group to which you want to move the device, and then click the **Move** button.

The device will be moved to the selected administration group. An appropriate message is displayed on the screen.

Moving a device to another administration group from the device details

To move a device to another administration group from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device to be moved.

   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device to be moved.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. In the **Select response actions** drop-down list, select **Move to group**.

   The **Move to group** window that opens on the right side of the screen displays the administration groups of the Administration Server that manages the selected device.

5. Select the administration group to which you want to move the device, and then click the **Move** button.

The device will be moved to the selected administration group. An appropriate message is displayed on the screen.

Moving a device to another administration group from an investigation graph

> This option is available if the investigation graph is built.

To move a device to another administration group from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device to be moved.

2. Click the **View on graph** button.

3. In the investigation graph that opens, click the device name to open the device details.

4. In the **Select response actions** drop-down list, select **Move to group**.

   The **Move to group** window that opens on the right side of the screen displays the administration groups of the Administration Server that manages the selected device.

5. Select the administration group to which you want to move the device, and then click the **Move** button.

The device will be moved to the selected administration group. An appropriate message is displayed on the screen.

Updating databases

To detect threats quickly and keep the protection level of a client device up to date, you have to regularly update databases and application modules on the device.

You can update databases on a device in one of the following ways:

- From the alert or incident details
- From the device details
- From an investigation graph

You can also configure the response action to run automatically when creating or editing a playbook.

To update databases on a device, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

It might take up to 15 minutes to launch a response action due to the synchronization interval between the managed device and Administration Server.

Updating databases from the alert or incident details

To update databases on a device from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device on which databases are to be updated.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which databases are to be updated.

2. In the window that opens, go to the **Assets** tab.

3. Select check box next to the devices on which databases are to be updated.

   You can select several devices, if necessary.

4. In the **Select response actions** drop-down list, select **Update databases**.

If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Updating databases from the device details

To update databases on a device from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device on which databases are to be updated.

   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which databases are to be updated.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. In the **Select response actions** drop-down list, select **Update databases**.

If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Updating databases from an investigation graph

This option is available if the investigation graph is built.

To update databases on a device from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which databases are to be updated.

2. Click the **View on graph** button.

3. In the investigation graph that opens, click the device name to open the device details.

4. In the **Select response actions** drop-down list, select **Update databases**.

If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Changing authorization status of devices

You can change an authorization status of a device when the analysis of an alert or incident shows that the protection level of the device is low or the device does harm to your infrastructure.

This response action is performed on devices with KICS for Networks installed.

You can change an authorization status of a device in one of the following ways:

- From the alert or incident details
- From the device details
- From a telemetry event
- From an investigation graph

You can also configure the response action to run automatically when creating or editing a playbook.

To change an authorization status of a device, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

Changing authorization status of devices from alert or incident details

To change an authorization status of a device from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device which authorization status is to be changed.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device which authorization status is to be changed.
2. In the window that opens, go to the **Assets** tab.
3. Select check box next to the device which authorization status is to be changed.

   You can select several devices, if necessary.
4. In the **Select response actions** drop-down list, select **Change authorization status**.
5. In the window that opens on the right side of the screen, select the new status of the device (*authorized* or *unauthorized*), and then click the **Change** button.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Changing authorization status of devices from the device details

To change an authorization status of a device from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device which authorization status is to be changed.

   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device which authorization status is to be changed.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. In the **Select response actions** drop-down list, select **Change authorization status**.

5. In the window that opens on the right side of the screen, select the new status of the device (*authorized* or *unauthorized*), and then click the **Change** button.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Changing authorization status of devices from a telemetry event

To change an authorization status of a device from a telemetry event:

1. In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device which authorization status is to be changed.

2. In the window that opens, go to the **Details** tab, and do one of the following:

   - Click the name of the required event and select the device.

   - Click the **Find in Threat hunting** button to go to the **Threat hunting** section and select the required device.

3. In the **Select response actions** drop-down list, select **Change authorization status**.

4. In the window that opens on the right side of the screen, select the new status of the device (*authorized* or *unauthorized*), and then click the **Change** button.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Changing authorization status of devices from an investigation graph

This option is available if the investigation graph is built.

To change an authorization status of a device from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device which authorization status is to be changed.

2. In the window that opens, click the **View on graph** button.

   The investigation graph opens.

3. Click the device name to open the device details.

4. In the **Select response actions** drop-down list, select **Change authorization status**.

5. In the window that opens on the right side of the screen, select the new status of the device (*authorized* or *unauthorized*), and then click the **Change** button.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The selected authorization status of the device in displayed in the alert or incident card, on the **Assets** tab → **Authorization status** column.

Moving files to quarantine

To prevent a threat distribution, you can move an infected file to quarantine in one of the following ways:

- From the alert or incident details
- From the device details
- From a telemetry event

  You can open the event details from alert or incident details, or from the **Threat hunting** section.
- From an investigation graph
- From the table of assets
- From the **Response history** section

You can also configure the response action to run automatically when creating or editing a playbook.

> To move the file to quarantine, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

> It might take up to 15 minutes to launch a response action due to the synchronization interval between the managed device and Administration Server.

Responding from the alert or incident details

To move a file to quarantine from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to the **Monitoring** → **Alerts** section. In the **ID** column, click the ID of the alert that includes the device on which the file to be moved to quarantine is located.

   - In the main menu, go to the **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which the file to be moved to quarantine is located.

2. In the window that opens, go to the **Assets** tab.

   > You can also go to the **Observables** tab, select check box next to the file that you want to move to quarantine, and then click the **Move to quarantine** button.

3. Select check box next to the device on which the file to be moved to quarantine is located.

   You can select several devices, if necessary.

4. In the **Select response actions** drop-down list, select **Move to quarantine**.

5. In the right pane that opens, do the following:

   a. In the **Full path** field, specify the full path to the file that you want to move to quarantine.

   b. In the **Assets** field, select the devices on which the file to be moved to quarantine is located.

   If at step 2 you selected the **Assets** tab, the names of the devices whose check boxes you selected are specified in the **Assets** field by default. You can change them and specify the devices that are not related to the alert or incident, but belong to the alert or incident tenant and its child tenants.

   > Only the assets on which you can perform the response action are available for selection.
   > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   c. If necessary, configure the following settings in the corresponding fields:

   - Specify either the SHA256 or MD5 hash of the file.

     > If at step 2 you selected the **Observables** tab, the hash is specified. You can change the value.

   - Write a description or a comment to the response action.

6. Click the **Move** button.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

Responding from the device details

To move a file to quarantine from the device details:

1. Do one of the following:

   - In the main menu, go to the **Monitoring** → **Alerts** section. In the **ID** column, click the ID of the alert that includes the device on which the file to be moved to quarantine is located.

   - In the main menu, go to the **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which the file to be moved to quarantine is located.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. Perform the same actions as described at steps 4–6 in the *Responding from the alert or incident details* procedure.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

Responding from a telemetry event

To move a file to quarantine from a telemetry event when opening it from the alert or incident details:

1. In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device on which the file to be moved to quarantine is located.

2. In the window that opens, go to the **Details** tab, and do one of the following:

   - Click the name of the required event and select the device.

   - Click the **Find in Threat Hunting** button to go to the **Threat hunting** section and select the required device.

   > You can also go to the **Observables** tab, select check box next to the file that you want to move to quarantine, and then click the **Move to quarantine** button.

3. In the **Select response actions** drop-down list, select **Move to quarantine**.

4. Perform the same actions as described at steps 4–6 in the *Responding from the alert or incident details* procedure.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

To respond from a telemetry event when opening it from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting.**

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select **Move to quarantine**.

4. Perform the same actions as described at steps 5–6 in *Responding from the alert or incident details* procedure.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

Responding from an investigation graph

This option is available if the investigation graph is built.

To move a file to quarantine from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the device on which the file to be moved to quarantine is located.

2. In the window that opens, click the **View on graph** button.

   The investigation graph opens.

3. Click the device name to open the device details.

4. In the **Select response actions** drop-down list, select **Move to quarantine**.

5. Perform the same actions as described at steps 4–6 in *Responding from the alert or incident details* procedure.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

Responding from the table of assets

To perform the response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   Archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select **Move to quarantine**.

4. In the right pane that opens, configure the response action settings as described at steps 5–6 in the *Responding from the alert or incident details* procedure.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

Responding from the Response history section

To perform the response action from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select **Move to quarantine**.

3. In the right pane that opens, configure the response action settings as described at steps 5–6 in the *Responding from the alert or incident details* procedure.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is deleted from the directory of the device where it is located and moved to the quarantine directory on the same device. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response actions in one of the following ways:

- From the **Response history** section in the main menu.

- From the **History** tab in the alert or incident details.

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, an **Error** message and information about the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Restoring files from quarantine

If you have reasons to believe that the file you moved to quarantine is safe, you can restore the file on the device in one of the following ways:

- From the **Quarantine** section of the main menu.

- From the results of the **Move to quarantine** response action.

You can also configure the response action to run automatically when creating or editing a playbook.

Responding from the Quarantine section

To respond from the Quarantine section:

1. In the main menu, go to **Operations** → **Repositories** → **Quarantine**.

2. Do one of the following:

   - Select the file to be restored from quarantine.

     You can select several files.

   - Click the name of the file to be restored from quarantine.

   The file details open.

3. Click the **Restore** button, and then confirm the operation.

   If necessary, in the **Description** field, you can write a comment before confirming the operation.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is restored on the device on which you previously moved it to quarantine and located on the original path (the one that you specified at step 5 in the Moving files to quarantine scenario). Otherwise, an error message is displayed.

Responding from the results of the **Move to quarantine** response action

To respond from the results of the Move to quarantine response action:

1. Open the result of the **Move to quarantine** response action in one of the following ways:

   - In the main menu, go to **Monitoring** → **Response history**.

   - In the main menu, go to **Monitoring**, and then select the **Alerts** or **Incidents** section. In the **ID** column, click the ID of the alert or incident that includes the asset or observable for which you performed the **Move to quarantine** response action. In the alert or incident details that open, move to the **History** tab.

2. For the result of the **Move to quarantine** response action, in the **Action status** column, click the **Success** link.

3. In the details window that opens, click the **Restore** button, and then confirm the operation.

4. If necessary, in the **Description** field, write a comment before confirming the operation.

If the operation is completed successfully, an appropriate message is displayed on the screen. The file is restored on the device on which you previously moved it to quarantine and located on the original path (the one that you specified at step 5 in Moving files to quarantine scenario). Otherwise, an error message is displayed.

Viewing the response action result

If Kaspersky Endpoint Security with a built-in agent is installed on your device, the result of the response action is displayed in the **Response history** section.

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, an **Error** message and information about the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Getting a file from quarantine

To perform the response action, you must have:

- One of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.
- Kaspersky Endpoint Security with a built-in agent installed on your devices.

The quarantined file located on a device can be obtained from the quarantine directory for further analysis. The obtained file will be displayed in the **Files** section of OSMP Console.

You can perform this response action in one of the following ways:

- From the **Quarantine** section of the main menu.
- From the results of the **Move to quarantine** response action.

You can also configure the response action to run automatically when creating or editing a playbook.

Responding from the Quarantine section

To respond from the Quarantine section:

1. In the main menu, go to **Operations** → **Repositories** → **Quarantine**.
2. Do one of the following:
   - Select the file that you want to get from quarantine.
   - Click the name of the file that you want to get from quarantine.

   The file details open.
3. Click the **Get file** button, and then confirm the operation.

   > If you want the file to be checked for threats after it is obtained, click the **Get & scan** button.

   If the operation is completed successfully, an appropriate message is displayed on the screen, and the quarantined file is obtained from the device. Otherwise, an error message is displayed.

Responding from the results of the Move to quarantine response action

To respond from the results of the Move to quarantine response action:

1. Open the result of the **Move to quarantine** response action in one of the following ways:
   - In the main menu, go to **Monitoring** → **Response history**.
   - In the main menu, go to **Monitoring**, and then select the **Alerts** or **Incidents** section. In the **ID** column, click the ID of the alert or incident that includes the asset or observable for which you performed the **Move to quarantine** response action. In the alert or incident details that open, move to the **History** tab.
2. For the result of the **Move to quarantine** response action, in the **Action status** column, click the **Success** link.
3. In the details window that opens, click the **Get file** button, and then confirm the operation.

   If you want the file to be checked for threats after it is obtained, click the **Get & scan** button.

Viewing the response action result

The result of the response action is displayed:

- In the **Files** section of the main menu.
- In the **Response history** section of the main menu.

  After you click the link in the **Action status** column, the file details open with the **Result** tab selected, where you can view and download a password-protected ZIP archive with the file to your computer, if necessary. The password is *infected*.

  If the response action was unsuccessful, an **Error** message and information about the reason of the error are displayed.

  If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Deleting files from quarantine

To perform the response action, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

You can delete an infected file from the quarantine directory on the device in one of the following ways:

- From the **Quarantine** section of the OSMP Console menu.
- From the results of the **Move to quarantine** response action.

You can also configure the response action to run automatically when creating or editing a playbook.

Responding from the Quarantine section

To respond from the Quarantine section:

1. In the main menu, go to **Operations** → **Repositories** → **Quarantine**.
2. Do one of the following:
   - Select the file to be deleted from quarantine.

     You can select several files.
   - Click the name of the file to be deleted from quarantine.

     The file details open.
3. Click the **Delete** button, and then confirm the operation.

   If the operation is completed successfully, an appropriate message is displayed on the screen. The file is permanently deleted from the quarantine directory on the device where it was located. Otherwise, an error message is displayed.

Responding from the results of the **Move to quarantine** response action

To respond from the results of the Move to quarantine response action:

1. Open the result of the **Move to quarantine** response action in one of the following ways:
   - In the main menu, go to **Monitoring** → **Response history**.
   - In the main menu, go to **Monitoring**, and then select the **Alerts** or **Incidents** section. In the **ID** column, click the ID of the alert or incident that includes the asset or observable for which you performed the **Move to quarantine** response action. In the alert or incident details that open, move to the **History** tab.
2. For the result of the **Move to quarantine** response action, in the **Action status** column, click the **Success** link.
3. In the details window that opens, click the **Delete** button, and then confirm the operation.

   If the operation is completed successfully, an appropriate message is displayed on the screen. The file is permanently deleted from the quarantine directory on the device where it was located. Otherwise, an error message is displayed.

Viewing the response action result

If Kaspersky Endpoint Security with a built-in agent is installed on your device, the result of the response action is displayed in the **Response history** section.

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, an **Error** message and information about the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Viewing information about KASAP users and changing learning groups

After configuring the integration between KASAP and KUMA, the following information from KASAP is available in OSMP Console when you view data about users associated with alerts or incidents:

- The learning group to which the user belongs.
- The learning courses completed by the user.
- The planned learning courses and their current progress.

You can view data about the KASAP user. To do this, you have to open a user details in one of the following ways:

- From the alert or incident details.
- From a telemetry event (if you open it from alert details).
- From an investigation graph.

  This option is available if the investigation graph is built.

To open a user details:

1. In the main menu, go to the **Monitoring** section, and then select the **Alerts** or **Incidents** section.

   If you want to open a user details from a telemetry event, select the **Alerts** section.

   If you want to open a user details from an investigation graph, select the **Incidents** section.

2. Click the ID of the required alert or incident.

3. In the window that opens, do one of the following:

   - If you want to open a user details from a telemetry event, go to the **Details** tab, and either click the name of the required event, and select the user; or click the **Find in Threat hunting** button to go to the **Threat Hunting** section, and then select the required user.
   - If you want to open a user details from alert or incident details, go to the **Assets** tab, and then click the name of the required user.
   - If you want to open a user details from investigation graph, click the **View on graph** button. In the investigation graph that opens, click the name of the required user.

   The **Account details** window opens on the right side of the screen.

4. Select the **Cybersecurity courses** tab.

   The window displays information about the KASAP user.

You can change the learning group of a KASAP user in one of the following ways:

- From the alert or incident details
- From a telemetry event (if you open it from alert details)
- From an investigation graph

  This option is available if the investigation graph is built.

You can also configure the response action to run automatically when creating or editing a playbook. In this case, if you move a user to the group for which the learning is not started, the user is not able to start learning.

To perform the response action, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

To change the KASAP user learning group:

1. In the main menu, go to the **Monitoring** section, and then select the **Alerts** or **Incidents** section.

   If you want to change the KASAP user learning group from a telemetry event, select the **Alerts** section.

   If you want to change the KASAP user learning group from an investigation graph, select the **Incidents** section.

2. Click the ID of the required alert or incident.

3. In the window that opens, do one of the following:

   - If you want to respond through a telemetry event, go to the **Details** tab, and either click the name of the required event, and then select the user; or click the **Find in Threat hunting** button to go to the **Threat hunting** section, and then select the required user.
   - If you want to respond through a user details, go to the **Assets** tab, and then click the name of the user.
   - If you want to respond through an investigation graph, click the **View on graph** button. In the investigation graph that opens, click the name of the user.

   The **Account details** window opens on the right side of the screen.

4. In the **Assign KASAP group** drop-down list, select the KASAP learning group to which you want to assign the user.

   Recalculation of the KASAP user training plan may take up to 30 minutes. It is not advisable to change the KASAP learning group during this period.

   The user is moved to the selected KASAP group. The KASAP company administrator receives a notification about the change in the learning group, and the study plan is recalculated for the selected learning group.

For details about learning groups and how to get started, refer to the KASAP documentation.

Enabling and disabling network isolation

You can enable and disable network isolation on a device in one of the following ways:

- From the alert or incident details
- From the device details
- From an investigation graph
- From the table of assets
- From the event details

  The option is only available to enable network isolation.

You can also configure the response action to run automatically when creating or editing a playbook.

> To perform the response actions you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

For the devices with Kaspersky Endpoint Security with a built-in agent installed, the network isolation status is displayed in the device details.

The device may have one of the following statuses: *Isolated*, *Isolating*, *Error applying isolation*, *Not isolated*.

Also, the network isolation status is displayed in the table of assets when you open it from the **Assets** tab in the alert or incident details or from **Asset management** → **Assets (Devices).** You can filter the values in the **Isolation status** column of the table to display the devices with the required status of network isolation.

Performing response actions from alert or incident details

To perform a response action from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.
2. In the window that opens, go to the **Assets** tab.
3. Select the select check box next to the required device.

   You can select several devices, if necessary.

4. In the **Select response actions** drop-down list, select **Network isolation**, and then in the context menu, select the response action that you want to perform:

   - **Enable**

     If you select this response action for a device on which network isolation is already enabled, the parameters are overwritten with new values.

     > Before enabling network isolation, ensure that the assets are imported from Kaspersky Security Center: in **Asset management** → **Tasks** → **Detection and response** the **KSC assets import** task is displayed with the **Completed** status.

   - **Disable**

     After you select this response action, in the confirmation dialog box, click **Disable**.

     You can select this response action for devices on which network isolation is enabled.

5. If you selected **Enable**, in the right pane that opens, configure the following settings:

   a. In the **Assets** drop-down list, specify the devices to be isolated.

   By default, the devices whose check boxes you previously selected are displayed. If necessary, you can change the devices. You cannot select the devices that have the following statuses: **Isolated**, **Isolating**, **Error applying isolation**.

   > Only the devices on which you can perform the response action are available for selection.
   > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   b. Specify the period of the device isolation and the units.

   c. In the **Description** field, write a description or a comment to the response action.

   d. If you want to add an exception from the network isolation rule, click the **Add exclusion** button, and in the **Network traffic direction** drop-down list, select one of the following values:

   - **Inbound**, and then in the corresponding fields specify the IP address of the device to be excluded from the network traffic blocking and a local ports range.
   - **Outbound**, and then in the corresponding fields specify the IP address of the device to be excluded from the network traffic blocking and the remote ports range.
   - **Inbound/Outbound**, and then specify the IP address of the device to be excluded from the network isolation.

6. Click the **Enable** button.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing response actions from the device details

To perform a response action from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.

   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. Perform the same actions as described at steps 4–6 in *Performing response actions from the device details*.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing response actions from an investigation graph

This option is available if the investigation graph is built.

To perform a response action from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.

2. In the window that opens, click the **View on graph** button.

   The investigation graph opens.

3. Click the device name to open the device details.

4. Perform the same actions as described at steps 4–6 in *Performing response actions from the device details*.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing response actions from the table of assets

To perform a response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

The archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select one of the following:

   - **Enable network isolation**, and then in the right pane that opens, configure the response action settings.

   - **Disable network isolation**, and then in the confirmation dialog box, click **Disable**.

Enabling network isolation from the event details

To enable network isolation from the event details:

1. In the main menu, go to **Monitoring** → **Threat hunting**.

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select **Enable network isolation**.

4. In the right pane that opens, configure the response action settings.

If you have not yet installed Kaspersky Endpoint Security with a built-in agent on your device, and encounter a failure when running the response actions, you have to make sure that the device name in Kaspersky Next XDR Expert is the same as in Kaspersky Anti Targeted Attack Platform.

Prevention rules

Prevention rules enable you to prevent files or processes that you consider insecure from running on the devices. Kaspersky Next XDR Expert identifies files based on their hash by using the MD5 or SHA256 hashing algorithms.

You can manage prevention rules in the following ways:

- Create and delete prevention rules by performing corresponding response actions

- Configure the response actions to run automatically when creating or editing a playbook

- Manually create, import, edit, and delete prevention rules from the table of prevention rules

Performing the Add prevention rule and Delete prevention rule response actions

Prevention rules enable you to prevent files or processes that you consider insecure from running on the devices. Kaspersky Next XDR Expert identifies files based on their hash by using the MD5 or SHA256 hashing algorithms.

You can configure the **Add prevention rule** and **Delete prevention rule** response actions to add and delete a prevention rule for a device or a file hash in one of the following ways:

- From the alert or incident details
- From the device details
- From an investigation graph
- From the table of assets
- From the **Response history** section
- From the event details

  This option is available for the **Add prevention rule** response action. You can open the event details from alert or incident details or from the **Threat hunting** section.
- From the file details

You can also configure the response action to run automatically when creating or editing a playbook.

> To perform the response actions you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

Performing response actions from alert or incident details

To perform a response action from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.
2. In the window that opens, go to the **Assets** tab.
3. Select the select check box next to the required device.

   You can select several devices, if necessary.
4. In the **Select response actions** drop-down list, select **Manage prevention rules**, and then in the context menu, select the response action that you want to perform:
   - **Add prevention rule**
   - **Delete prevention rule**

5. In the window that opens, set the necessary parameters for the response action you selected at step 4:

- **For adding prevention rule**

  In the right pane that opens, configure the following settings:

  1. In the **MD5 or SHA256** field, specify a hash of the file that you want to block.

     If you want to specify more than one hash, click the **Add hash** button.

  2. If necessary, configure the following settings:

     - Change the scope of the response action.

       By default, in the **Scope** group of settings the **Specific assets** option is selected, and the **Assets** drop-down list contains the names of the devices whose check box you previously selected.

       You can either change the devices for which the prevention rule is to be added, or select the **All assets** option in the **Scope** group of settings to add the prevention rule for all the devices that belong to the alert or incident tenant and its child tenants.

       > Only the assets on which you can perform the response action are available for selection. The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

     - Enable the **Notify user about blocking file execution** toggle switch.

     - In the **Description** field, write a description or a comment to the response action.

  3. Click the **Add** button.

     The window is closed.

- **For deleting prevention rule**

  In the right pane that opens, configure the following settings:

  1. Select what you want to delete.

     By default, in the **Rules** group of settings the **Specific rules** option is selected. If you want to delete a prevention rule by file hash, in the **MD5 or SHA256** field specify a hash of the file to be deleted. If you want to specify more than one hash, click the **Add hash** button.

     If you want to delete all prevention rules, select **All rules** option.

  2. If necessary, configure the following settings:

     - Change the scope of the response action.

       By default, in the **Scope** group of settings the **Specific assets** option is selected, and the **Assets** drop-down list contains the names of the devices whose check box you previously selected.

       You can either change the devices for which the prevention rule is to be deleted, or select the **All assets** option in the **Scope** group of settings to delete prevention rules for all the devices that belong to the alert or incident tenant and its child tenants.

Only the assets on which you can perform the response action are available for selection. The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

- In the **Description** field, write a description or a comment to the response action.

3. Click the **Delete** button.

   The window is closed.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing response actions from the device details

To perform a response action from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.

   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. Perform the same actions as described at steps 4-5 in *Performing response actions from the device details*.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing response actions from an investigation graph

This option is available if the investigation graph is built.

To perform a response action from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.

2. In the window that opens, click the **View on graph** button.

   The investigation graph opens.

3. Click the device name to open the device details.

4. Perform the same actions as described at steps 4-5 in *Performing response actions from the device details*.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing a response action from the table of assets

To perform a response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   > The archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select **Manage prevention rules**, and then in the context menu, select the response action that you want to perform.

4. In the right pane that opens, configure the response action settings as described at step 5 in *Performing a response action from the alert or incident details*.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing a response action from the Response history section

To perform a response action from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select either **Add prevention rule** or **Delete prevention rule**.

3. In the right pane that opens, configure the response action settings as described at step 5 in *Performing a response action from the alert or incident details*.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing a response action from the event details

> This option is available for the **Add prevention rule** response action.

To perform a response action from the event details when opening it from the alert or incident details:

1. In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.

2. In the window that opens, go to the **Details** tab, and select the required file hash.

3. Click the **Add prevention rule** button, and then select the device for which you want to add the prevention rule.

   You can also go to the **Observables** tab, select check box next to the file hash that you want to block, and then click **Add prevention rule**.

4. Perform the same actions as described at step 5 in *Performing response actions from the device details*.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

To perform a response action from the event details when opening it from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting.**

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select **Add prevention rule**.

4. Perform the same actions as described at step 5 in *Performing response actions from the device details*.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response action in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

After you click the link in the **Action status** column, the window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and the **View asset** button are displayed. You can click the button to view the status of the prevention rule on the device.

Otherwise, the **Error** message and the information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

If you have not yet installed Kaspersky Endpoint Security with a built-in agent on your device, and encounter a failure when running the response actions, you have to make sure that the device name in Kaspersky Next XDR Expert is the same as in Kaspersky Anti Targeted Attack Platform.

Working with the table of prevention rules

Prevention rules enable you to prevent files or processes that you consider insecure from running on the devices. Kaspersky Next XDR Expert identifies files based on their hash by using the MD5 or SHA256 hashing algorithms.

You can manage prevention rules in the following ways:

- Manually create, import, edit, and delete prevention rules from the table of prevention rules, as described in this article
- Create and delete prevention rules by performing corresponding response actions
- Configure the response actions to run automatically when creating or editing a playbook

> To open the table of prevention rules, you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator, SOC manager, Tier 1 analyst, Tier 2 analyst, Junior analyst, Approver, Auditor. To create or import prevention rules, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

To open the table of prevention rules, in the main menu, go to **Monitoring** → **Resources and services** → **Prevention rules**.

The table contains the following columns:

- **Name**. For manually created prevention rules, the column displays the name that you specified in the **Create prevention rule** pane. For imported prevention rules, the file hash is displayed in the column.
- **Hash**. The MD5 or SHA256 hash of the file to be blocked.
- **Tenant owner**. The tenant to which the prevention rule belongs.

  In the distributed solution mode, if a tenant was created on a secondary node, the value in the **Tenant** filed is displayed with the secondary node icon (⊟).
- **Assets**. The devices for which the prevention rule is applied. The column either displays the number of devices or the **All assets of tenant** value.
- **Created on**. The date and time of the prevention rule creation or import.
- **Created by**. The user who created or imported the prevention rule.
- **Updated on**. The date and time when the prevention rule settings were last edited.
- **Updated by**. The user who last edited the prevention rule settings.
- **Rule status**. If the prevention rule is enabled, the column displays the **Enabled** value for the rule. Otherwise, the **Disabled** value is displayed.
- **Notify on file block**. If when creating or editing the prevention rule you disabled notification about blocking a file, the column displays the **No** value. Otherwise, the **Yes** value is displayed.

In the table you can perform the following actions:

- Create or import a prevention rule.
- Delete a prevention rule by selecting a check box next to the required rule name, and then clicking the **Delete** button in the toolbar.
- View and edit the prevention rule details by clicking the rule name in the table.

  The **Edit prevention rule** pane opens with the **Details** tab selected. You can edit the prevention rule settings. The settings are the same as when you create a prevention rule from the table of rules. Also, you can click the **Go to Threat hunting** button to view the events obtained when a prevention rule was triggered.

  On the **Change log** tab, you can view all the changes applied to the prevention rule: manually from the table of prevention rules, by performing response actions, or by launching a playbook.

Creating a prevention rule

To create a prevention rule:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Prevention rules**.

   The table of prevention rules is displayed.

2. In the toolbar, click the **Create** button.

3. In the **Create prevention rule** pane that opens, configure the settings:

   a. In the **Tenant owner** field, specify a tenant to which the prevention rule will belong.

   > After you save the prevention rule settings, you cannot edit the value in the **Tenant owner** field. If you want to change the tenant, you have to delete the prevention rule, create a new one, and specify the required value in the **Tenant owner** field.

   b. In the **MD5 or SHA256** field, specify a hash of the file that you want to block.

   If you want to specify more than one hash, click the **Add hash** button.

   > The application checks, if the hash is present in the tenant in the list of file hashes to be blocked. You cannot specify such hashes when creating a new prevention rule.

   c. If necessary, configure the following settings:

   - Change the scope of the prevention rule.

     By default, in the **Apply rule to** group of settings, the **All assets** option is selected. This means the prevention rule is applied to all devices. You can select the **Specific assets** option, and then in the **Assets** field that displays, specify one or several devices to which you want to apply the prevention rule.

     > If you set the devices for which you want to apply the prevention rule, and then a device is deleted, such device will be deleted from the prevention rule as well, and a corresponding information will be displayed in the **Change log** tab of the prevention rule details.

- In the **Rule name** field, specify the name of the prevention rule. If you do not specify the name, the field is displayed empty.
- In the **Notify user about blocking file execution** field, disable the toggle switch if you do not want to get notifications on file block.

  By default, the option is enabled.
- In the **State** field, disable the toggle switch if you do not want the prevention rule to be applied.

  If you want to create a prevention rule with the **Disabled** state, you have to configure the other settings first. Otherwise, you will not be able to save the rule.

d. Click the **Create** button.

The pane is closed, and the prevention rule is displayed in the list of rules.

Importing a prevention rule

To import a prevention rule:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Prevention rules**.

   The table of prevention rules is displayed.
2. In the toolbar, click the **Import** button.

   Prevention rules imported from a file will be applied to the assets that belong to the tenant including the child tenants.
3. In the **Import prevention rules** pane that opens, configure the necessary settings:

   a. Select a CSV file from your computer.

   You can add only one CSV file that contains hashes. The number of hashes equals the number of prevention rules to be created. The maximum file size must not exceed 15 MB

   b. In the **Tenant** drop-down list, select the tenant to which the prevention rules are to be applied.

   By default, the tenant to which you have access rights is selected. If necessary, you can change the value.

   If you have access rights to several tenants, the value is empty, and you must specify the required tenant.

   c. Click the **Import** button.

The pane is closed, and the imported prevention rule is displayed in the table of rules.

Running a program

You can perform the response action in one of the following ways:

- From the alert or incident details
- From the device details

- From an investigation graph
- From the table of assets
- From the **Response history** section
- From the event details

You can also configure the response action to run automatically when creating or editing a playbook.

> To perform the response actions you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

Performing the response action from alert or incident details

To perform the response action from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.
2. In the window that opens, go to the **Assets** tab.
3. Select the select check box next to the required device.

   You can select several devices, if necessary.
4. In the **Select response actions** drop-down list, select **Run**, and then in the context menu, click **Run program**.
5. In the right pane that opens, do the following:

   a. In the **Full path** field, specify the full path to the executable file to be run.

   b. If necessary, configure the following settings:

   - Change the scope of the response action.

     By default, in the **Scope** drop-down list the **Specific assets** option is selected, and the **Assets** drop-down list contains the names of the devices whose check box you previously selected. It means, the executable file is to be run on these devices.

     You can either change the devices or select the **All assets** option in the **Scope** drop-down list to run executable file on all the devices that belong to the alert or incident tenant (and its child tenants).

Only the assets on which you can perform the response action are available for selection.
The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

- In the **Command line parameters** field, specify additional options for running the executable file, for example: `/all argument`.
- In the **Working directory** field, specify the working directory.
- In the **Description** field, write a description or a comment to the response action.

6. Click the **Run** button.

   The pane is closed. The window is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing the response action from the device details

To perform the response action from the device details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.
   - In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.
2. In the window that opens, go to the **Assets** tab.
3. Click the name of the required device, and then in the drop-down list, select **View properties**.
4. Perform the same actions as described at steps 4–6 in *Performing a response action from the device details* procedure.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing the response action from an investigation graph

This option is available if the investigation graph is built.

To perform the response action from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.
2. In the window that opens, click the **View on graph** button.

   The investigation graph opens.

3. Click the device name to open the device details.

4. Perform the same actions as described at steps 4–6 in *Performing response actions from the device details* procedure.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

> If you have not yet installed Kaspersky Endpoint Security with a built-in agent on your device, and encounter a failure when running the response actions, you have to make sure that the device name in Kaspersky Next XDR Expert is the same as in Kaspersky Anti Targeted Attack Platform.

Performing the response actions from the table of assets

To perform the response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   > The archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select **Run program**.

4. In the right pane that opens, configure the response action settings as described at steps 4–6 in *Performing a response action from the device details* procedure.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing the response action from the Response history section

To perform the response action from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select **Run program**.

3. In the right pane that opens, configure the response action settings as described at steps 4–6 in *Performing a response action from the device details* procedure.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Performing the response action from the event details

To perform the response action from the event details:

1. In the main menu, go to **Monitoring** → **Threat hunting.**

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select **Run program**.

4. In the right pane that opens, configure the response action settings as described at steps 4–6 in *Performing a response action from the device details* procedure.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response actions in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

After you click the link in the **Action status** column, the window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, the **Error** message and the information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Managing a service

You can remotely start, stop, pause, resume, or remove a service on a device.

To perform the **Manage service** response action, you must have:

- One of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst
- Kaspersky Endpoint Security with a built-in agent installed on your devices

You can perform the response actions in one of the following ways:

- From the alert or incident details
- From the device details
- From a telemetry event

  You can open the event details from the alert or incident details or from the **Threat hunting** section.

- From an investigation graph

  This option is available if the investigation graph is built.

- From the table of assets
- From the **Response history** section

You can also configure the response action to run automatically when creating or editing a playbook.

Responding from the alert or incident details

To respond from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.
   - In the main menu, go to the **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.
2. In the window that opens, go to the **Assets** tab.

   You can also go to the **Observables** tab, select the check box next to the alert for which you want to perform the response action, and then click **Manage service**.

3. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

4. In the **Select response actions** drop-down list, select **Manage service**.
5. In the right pane that opens, configure the following settings in the corresponding fields:

   a. The devices on which you want to manage the service.

   By default, in the **Scope** group of settings, the **Specific assets** option is selected, and the **Assets** drop-down list contains the names of the devices whose check box you selected at step 3.

   You can either change the devices or select the **All assets** option in the **Scope** group of settings to manage service on all the devices that belong to the alert or incident tenant and its child tenants.

   Only the assets on which you can perform the response action are available for selection.
   The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   b. The service name.

   c. The action to be performed on the service: **Modify startup type**, **Start**, **Stop**, **Pause**, **Resume**, or **Delete**.

If you select the **Modify startup type** action, you have to select the value in the **New startup type** drop-down list.

Stopping, pausing, deleting services, or changing the start type of services that affect the functioning on the host is strongly not recommended.
List of services for which management is not recommended

- AVP*
- BFE
- CertPropSvc
- CoreMessagingRegistrar
- CryptSvc
- DcomLaunch
- Dhcp
- DispBrokerDesktopSvc
- EventLog
- EventSystem
- LanmanServer
- LanmanWorkstation
- LSM
- mpssvc
- Netlogon
- Netman
- PlugPlay
- PolicyAgent
- Power
- ProfSvc
- RasMan
- RpcEptMapper
- RpcSs

- SamSs
- SCardSvr
- SecurityHealthService
- SessionEnv
- Spooler
- SstpSvc
- TermService
- UmRdpService
- UserManager
- UsoSvc
- VaultSvc
- W32Time
- WebClient
- Winmgmt
- WlanSvc
- wscsvc

d. If necessary, configure the following settings:

- Specify either the SHA256 or MD5 hash of the file.

  If at step 2 you selected the **Observables** tab, the hash is specified. You can change the value.

- Write a description or a comment to the response action.

6. Click the **Launch** button.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the device details

To respond from the device details:

1. Do one of the following:

   - In the main menu, go to the **Monitoring** → **Alerts** section. In the **ID** column, click the ID of the alert that includes the required device.

   - In the main menu, go to the **Monitoring** → **Incidents** section. In the **ID** column, click the ID of the incident that includes the required device.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

   The device details pane opens.

4. Perform the same actions as described at steps 4–6 in *Responding from the alert or incident details* procedure.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from a telemetry event

To respond from a telemetry event when opening it from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the required device.

   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the required device.

2. In the window that opens, go to the **Details** tab, and do one of the following:

   - Click the name of the required event and select the device.

   - Click the **Find in Threat hunting** button to go to the **Threat hunting** section, and then select the required device.

3. Perform the same actions as described at steps 4–6 in the *Responding from the alert or incident details* procedure.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

To respond from a telemetry event when opening it from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting**.

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details pane that opens, click the device name, and then from the drop-down list, select **Manage service**.

4. In the right pane that opens, configure the response action settings as described at step 5 in the *Responding from the alert or incident details* procedure, and then launch the response action.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from an investigation graph

To respond from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the required device.

2. Click the **View on graph** button.

3. In the investigation graph that opens, click the device name, and then in the drop-down list, select **View properties**.

4. In the device details pane that open, perform the same actions as described at steps 4–6 in *Responding from the alert or incident details* procedure.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the table of assets

To respond from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   > Archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select **Manage service**.

4. In the right pane that opens, configure the response action settings as described at steps 5–6 in the *Responding from the alert or incident details* procedure.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the Response history section

To respond from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select **Manage service**.

3. In the right pane that opens, configure the response action settings as described at step 5 in *Responding from the alert or incident details* procedure, and then launch the response action.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response action in one of the following ways:

- From the **Response history** section in the main menu.
- From the **History** tab in the alert or incident details.

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, an **Error** message and information about the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Blocking IP addresses through Kaspersky NGFW

You can prevent a potential threat distribution and malicious activities on your networked devices through Kaspersky NGFW by adding IP addresses to the blocklist.

You can also configure the response action to run automatically when creating or editing a playbook.

> The response action is available if the Open Single Management Platform distribution kit includes the Kaspersky NGFW web plug-in.
> To perform the response action through Kaspersky NGFW, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

To perform the response action through Kaspersky NGFW:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the link with the alert ID you need.
   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.

2. In the window that opens, go to the **Observables** tab.

3. Select the **HostIP** check box.

4. In the **Select response action** drop-down list, select the **Add IP to blocklist on network devices** response action.

The selected response action is completed. An appropriate message is displayed on the screen.

Getting files for further analysis

This section contains response actions aimed to obtain the evidence of a security breach, for further investigation. You can get:

- File
- Registry key
- NTFS metafiles
- Forensics
- Disk image
- Memory dump
- Process memory dump

To perform the response actions, you must have:

- One of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.
- Kaspersky Endpoint Security with a built-in agent installed on your devices.

You can perform the response actions in one of the following ways:

- From the alert or incident details
- From the device details
- From a telemetry event

  You can open the event details from alert or incident details, or from the **Threat hunting** section.
- From an investigation graph

  > This option is available if the investigation graph is built.
- From the table of assets
- From **Response history** section

You can also configure the response action to run automatically when creating or editing a playbook.

Responding from the alert or incident details

To perform a response action from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the link with the alert ID you need.
   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.
2. In the window that opens, go to the **Assets** tab.
3. Select the check box next to the device for which you want to perform a response action.
   If you want to perform the **Registry key** or **Forensics** response action, you can select several devices.
4. In the **Select response actions** drop-down list, click **Get**.
5. In the context menu that opens, select the required response action.

Responding from the device details

To perform a response action from the device details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the link with the alert ID you need.
   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.
2. In the window that opens, go to the **Assets** tab.
3. Click the name of the required device, and then in the drop-down list, select **View properties**.
   The device details window opens.
4. In the **Select response actions** drop-down list, click **Get**.
5. In the context menu that opens, select the required response action.

Responding from a telemetry event

To perform a response action from a telemetry event when opening it from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the link with the alert ID you need.
   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.

2. In the window that opens, go to the **Details** tab, and do one of the following:
   - Click the name of the required event and select the device.
   - Click the **Find in Threat hunting** button to go to the **Threat hunting** section and select the required device.
3. In the **Select response actions** drop-down list, click **Get**.
4. In the context menu that opens, select the required response action.

To respond from a telemetry event when opening it from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting.**

   The table of events opens.
2. Click the name of the event whose details you want to open.
3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select the required response action.

Responding from an investigation graph

To perform a response action from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**. In the ID column, click the ID of the incident that includes the device for which you want to perform a response action.
2. In the window that opens, click the **View on graph** button.

   The investigation graph opens.
3. Click the name of the required device, and then in the drop-down list, select **View properties**.

   The device details window opens.
4. In the **Select response actions** drop-down list, click **Get**.
5. In the context menu that opens, select the required response action.

Responding from the table of assets

To perform the response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.
2. Select the check box next to the device for which you want to perform the response action.

   If you want to perform the Forensics or Registry key response actions, you can select several devices.

   > Archived devices cannot be selected.
3. In the **Select response actions** drop-down list, select the required response action.

Responding from the Response history section

To perform the response action from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select the required response action.

Getting a file

To get a file:

1. Select the **File** response action.

   The **Get file** pane opens.

2. In the **Full path** field, specify the path to the file that you want to obtain.

3. If necessary, configure the following settings:

   - In the **Assets** drop-down list, change the device from which you want to obtain a file.

   - By default, the field contains the name of the device whose check box you previously selected. You can select devices that are not related to the alert or incident, but belong to the alert or incident tenant and its child tenants.

     > Only the assets on which you can perform the response action are available for selection.
     > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   - In the **MD5 or SHA256** field, specify the hash of the file that you want to obtain.

   - If you do not want to scan the file, disable the **Scan file** option.

     By default, the option is enabled. This means that after a file is obtained, it is checked for threats.

     > When launching the response action within a playbook, you have to add this option manually if you want to scan the obtained file.

   - In the **Description** field, write a description or a comment to the response action.

4. Click the **Get** button.

   The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, the file is saved to a ZIP archive. The password is *infected*.

Viewing the response action result

You can view and download the archive in one of the following ways:

- From the **Response history** section in the main menu, or from the **History** tab in the alert or incident details.
- From the **Files** in the main menu, or from the **Files** tab in the alert or incident details.

If at step 2 you did not disable the **Scan file** option, the result of the file scan is also displayed. After you click the link in the **Action status** column, the file details open with the **Result** tab selected, where you can view the scan results and download the file to your computer, if necessary.

Getting a registry key

To get a registry key:

1. Select the **Registry key** response action.

   The **Get registry key** pane opens.

2. In the **Relative path to the registry key** field, specify the path to the registry key that you want to obtain in one of the following formats:

   - Relative to the root key, for example:

     \REGISTRY\MACHINE\SOFTWARE\Microsoft\WindowsUpdate\Orchestrator

   - Relative with the full name of the root key, for example:

     HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsUpdate\Orchestrator

   - Relative with an abbreviation instead of the full name of the root key, for example:

     HKLM\SOFTWARE\Microsoft\WindowsUpdate\Orchestrator

3. If necessary, configure the following settings:

   - In the **Assets** drop-down list, change the devices from which you want to get registry keys.

     By default, the field contains the names of the devices whose check boxes you previously selected. You can select devices that are not related to the alert or incident, but belong to the alert or incident tenant and its child tenants.

     > Only the assets on which you can perform the response action are available for selection.
     > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

4. In the **Description** field, write a description or a comment to the response action.

5. Click the **Get** button.

The pane is closed. If the response action is completed successfully, the appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, an REG file is generated. This file contains the list of registry keys and their values, located at the path that you specified.

Viewing the response action results

You can view and download the file in one of the following ways:

- From the **Files** section in the main menu, or from the **Files** tab in the alert or incident details
- From the **Response history** section in the main menu, or from the **History** tab in the alert or incident details

  After you click the link in the **Action status** column, the window opens with the **Result** tab selected. If the response action completed successfully, a **Success** message and the generated file are displayed. Otherwise, an **Error** message and the information about the reason of the error are displayed.

  If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Getting NTFS metafiles

> The **Get NTFS metafiles** task retrieves only the default data stream of a file. Alternate (named) data streams are not supported.

To get NTFS metafiles:

1. Select the **NTFS metafiles** response action.

   The **Get NTFS metafiles** pane opens.

2. In the **Metafiles** field, select the check boxes next to the metafiles that you want to get:

   - $MFT
   - $LogFile
   - $Boot
   - $Secure
   - $Extend/$UsnJrnl

3. In the **Volume** field, specify the name of the disk from which you want to get metafiles.

   By default, the *%SystemDrive%* value is specified.

4. If necessary, configure the following settings:

   - In the **Assets** drop-down list, select the device from which you want to get metafiles.

     By default, the field contains the name of the device whose check box you previously selected. You can select a device that is not related to the alert or incident, but belongs to the alert, or incident tenant and its child tenants.

Only the assets on which you can perform the response action are available for selection.
The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

- In the **Description** field, you can write a description or a comment to the response action.

5. Click the **Get** button.

   The pane is closed. If the response action is completed successfully, the appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, the NTFS metafiles are saved to a ZIP archive.

Viewing the response action result

You can view and download the archive in one of the following ways:

- From the **Files** section in the main menu, or from the **Files** tab in the alert or incident details
- From the **Response history** section in the main menu, or from the **History** tab in the alert or incident details

  After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and the obtained files are displayed. Otherwise, an **Error** message and the information about the reason of the error are displayed.

  If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Getting forensics

To get forensics:

1. Select the **Forensics** response action.

   The **Get forensics** pane opens.

2. In the **Information type** section, select the check boxes next to the type of data that you want to obtain:

   - **Processes list**. For a list of processes running on the device.
   - **Autorun points list**. For a list of autorun points.

     The list includes information about applications added to the startup folder or registered in the Run registry keys, as well as applications that are automatically run when the specified device is started and a user logs in to the operating system on the device.

     List of supported autorun points

     The following types of autorun points are supported:

     - Logon
     - Shell
     - Office

- Browsers
- Scheduler
- Services
- Drivers
- Cryptography
- Debuggers
- COM
- SessionManager
- Network
- LSA
- Applications
- WMI
- GroupPolicy
- DriverLog
- BootLog
- Process
- OsUpdate
- Printer
- Unspecified
- OsLoader
- EfiLoader

- **File list**. For a list of files and folders stored on the disk.

The maximum number of files displayed in the list is 20,000.

3. If you selected the **File list** check box, do the following:

   a. Specify the scope of the files search by selecting one of the following group of settings:

      - **All local discs**. The list will include the files stored in all folders on local disks.
      - **Folder**, and then in the **Start folder** field, specify the path to the folder from which the file search must start.

        The list will include the files stored in the specified folder and its subfolders.

   b. If necessary, specify the following search criteria for files in folders:

      - The mask of files to be included in the list of files.
      - The **Alternate data streams** option to record information about alternate data streams in the file list.

        The toggle switch is enabled by default.
      - Maximum nesting level of folders in which the application searches for files.
      - In the **Exclusions** field, specify the path to the folder in which you want to prohibit the search for information about files.

4. If necessary, configure the following settings:

   - In the **Assets** drop-down list, select the devices from which you want to get forensics.

     By default, the field contains the names of the devices whose check boxes you previously selected. You can select devices that are not related to the alert or incident, but belong to the alert, or incident tenant and its child tenants.

     > Only the assets on which you can perform the response action are available for selection.
     > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   - In the **Description** field, write a description or a comment to the response action.

5. Click the **Get** button.

   The pane is closed. If the response action is completed successfully, the appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, the selected data are saved to a password-protected ZIP archive. The password is *infected*.

Viewing the response action result

You can view and download the archive in one of the following ways:

- From the **Files** section in the main menu, or from the **Files** tab in the alert or incident details.
- From the **Response history** section in the main menu, or from the **History** tab in the alert or incident details.

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and the obtained file are displayed. Otherwise, an **Error** message and the information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Getting a disk image

To get a disk image:

1. Select the **Disk image** response action.

   The **Get disk image** pane opens.

2. In the **Path to a shared network resource** field, specify the path to the shared network resource, in the UNC format.

3. In the **User name** field, select the user name of the account to be used for accessing the shared network resource.

   > You have to configure the credentials for accessing the shared network resource in advance in the tenant settings.

4. In the **Disk type** group of settings, select one of the following:

   - **Logical**, and then in the **Volume** field, specify the drive letter without the colon and slash, for example, E or C.

     The default value is %SystemDrive%.

   - **Physical**, and then in the **Physical drive** field, specify the path to the disk in the following format:

     - For a device with Kaspersky Endpoint Security for Windows: \\.\PHYSICALDRIVE0, \\.\PHYSICALDRIVE1, \\.\PHYSICALDRIVE2

     - For a device with Kaspersky Endpoint Security for Linux: /dev/sda

5. If you want to divide the disk image file into several parts, turn on the **Split file into parts** option, and then specify the minimum size of one part of the saved file in GB.

6. If necessary, configure the following settings:

   - In the **Assets** drop-down list, change the device from which you want to get a disk image.

     By default, the field contains the name of the device whose check box you previously selected. You can select devices that are not related to the alert or incident, but belong to the alert, or incident tenant and its child tenants.

Only the assets on which you can perform the response action are available for selection.
The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

- In the **Description** field, write a description or a comment to the response action.

7. Click the **Get** button.

   The pane is closed. If the response action is completed successfully, the appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, the disk image is saved to a EWF file.

The files cannot be overwritten: if the destination folder already includes a file with the name, the name of your EWF file will contain a postfix, for example, Filename-1.

Viewing the response action result

You can view the result of the response actions in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and the path to the file with the memory disk image located in the shared network resource are displayed. You can copy the path.

Otherwise, an **Error** message and information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

You cannot move to the shared network resource directly from the response action result.

Getting a memory dump

To get a memory dump:

1. Select the **Memory dump** response action.

   The **Get memory dump** pane opens.

2. In the **Path to a shared network resource** field, specify the path to the shared network resource, in the UNC format.

3. In the **User name** field, select the user name of the account to be used for accessing the shared network resource.

   You have to configure the credentials for accessing the shared network resource in advance in the tenant settings.

4. If necessary, configure the following settings:

   - In the **Assets** drop-down list, select the device from which you want to obtain the memory dump.

     By default, the field contains the name of the device whose check box you previously selected. You can select devices that are not related to the alert or incident, but belong to the alert, or incident tenant and its child tenants.

     Only the assets on which you can perform the response action are available for selection.
     The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   - In the **Description** field, write a description or a comment to the response action.

5. Click the **Get** button.

   The pane is closed. If the response action is completed successfully, the appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, the disk image is saved to a RAW file.

The files cannot be overwritten: if the destination folder already includes a file with the name, the name of your RAW file will contain a postfix, for example, Filename-1.

Viewing the response action result

You can view the result of the response actions in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and the path to the file with the memory dump located in the shared network resource are displayed. You can copy the path.

Otherwise, an **Error** message and information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

You cannot move to the shared network resource directly from the response action result.

Getting a process memory dump

To get a process memory dump:

1. Select the **Process memory dump** response action.

   The **Get process memory dump** pane opens.

2. In the **Process ID** field, specify the ID of the process for which you want to get a memory dump.

3. If necessary, configure the following settings:

   - In the **Assets** drop-down list, change the device from which you want to obtain the process memory dump.

     By default, the field contains the name of the device whose check box you previously selected. You can select devices that are not related to the alert or incident, but belong to the alert, or incident tenant and its child tenants.

     > Only the assets on which you can perform the response action are available for selection.
     > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   - In the **MD5 or SHA256** field, specify the hash of the file for whose process you want to get a memory dump.

   - In the **Description** field, write a description or a comment to the response action.

4. Click the **Get** button.

   The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

As a result of the response action, the memory dump of the process is saved to a ZIP archive.

Viewing the response action result

You can view and download the archive in one of the following ways:

- From **Files** in the main menu, or from the **Files** tab in the alert or incident details

- From the **Response history** section in the main menu, or from the **History** tab in the alert or incident details

  After you click the link in the **Action status** column, a window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and a password-protected ZIP archive with the scan result are displayed. You can download the archive to your computer. The password is the following: infected.

  Otherwise, the **Error** message and the information about the reason of the error are displayed.

  If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

Scanning devices for threats

To prevent a threat distribution on an infected device, you can scan the device for threats by performing one of the following response actions:

- Run malware scan
- YARA scan
- IOC scan

> To perform the response actions, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

You can perform the response actions in one of the following ways:

- From the alert or incident details
- From the device details
- From a telemetry event

  You can open the event details from alert or incident details, or from the **Threat hunting** section.
- From an investigation graph

  > This option is available if the investigation graph is built.
- From the table of assets
- From the **Response history** section
- From the list of YARA rules or IOC rules

  > This option is available for YARA scan and IOC scan response actions.

You can also configure the response action to run automatically when creating or editing a playbook.

Running a scan from the alert or incident details

To scan a device from the alert or incident details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device to be scanned.
   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device to be scanned.
2. In the window that opens, go to the **Assets** tab.
3. Select the check box next to the device to be scanned.

   You can select several devices, if necessary.
4. In the **Select response actions** drop-down list, click **Scan**.
5. In the context menu that opens, select the required response action.

Running a scan from the device details

To scan a device from the device details:

1. Do one of the following:
   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device to be scanned.
   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device to be scanned.
2. In the window that opens, go to the **Assets** tab.
3. Click the name of the required device, and then in the drop-down list, select **View properties**.
4. In the **Select response actions** drop-down list, click **Scan**.
5. In the context menu that opens, select the required response action.

Running a scan from an investigation graph

To scan a device from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device to be scanned.
2. Click the **View on graph** button.
3. In the investigation graph that opens, click the device name to open the device details.

4. In the **Select response actions** drop-down list, click **Scan**.

5. In the context menu that opens, select the required response action.

Running a scan from a telemetry event

To scan a device from a telemetry event when opening it from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the link with the alert ID you need.

   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the link with the incident ID you need.

2. In the window that opens, go to the **Details** tab, and do one of the following:

   - Click the name of the required event, and then select the device.

   - Click the Find in Threat hunting button to go to the **Threat hunting** section, and then select the required device.

3. In the **Select response actions** drop-down list, click **Scan**.

4. In the context menu that opens, select the required response action.

To scan a device from a telemetry event when opening it from Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting.**

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details window that opens, click the device name, and then from the drop-down list that opens, select the required response action.

Running a scan from the table of assets

To run a scan from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   > The archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select the required response action.

Running a scan from the Response history section

To run a scan from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.
   The table that contains the response history for all alerts and incidents opens.
2. In the **Response** drop-down list, select the required response action.

Running a scan from the list of YARA rules or IOC rules

To scan a device from the list of YARA rules or IOC rules:

1. In the main menu, go to **Rules and services**, and then select either **YARA rules** or **IOC rules**.
   The list of rules is displayed.
2. Select the check box next to one or several devices for which you want to run the scan.
3. In the toolbar, click the button:
   - **Run YARA scan** if at step 1 you selected **YARA rules**.
   - **Run IOC scan** if at step 1 you selected **IOC rules**.
4. In the right pane that opens, configure the scan settings.

Running a malware scan

> It might take up to 15 minutes to launch a response action due to the synchronization interval between the managed device and Administration Server.

If necessary, you can click the **Edit in KUMA** button to edit parameters of the device in KUMA Console.

If you selected the **Run malware scan** response action, in the **Malware scan** pane do the following:

1. Select the type of malware scan:
   - **Full scan**
     You can switch the **Scan network drives** toggle button to include network devices into the scan. By default, this option is disabled.

     > A full scan can slow down the device due to an increased load on its operation system.
     > For devices with 12.4 version of Kaspersky Endpoint Security for Linux installed, full scan is not supported.

   - **Critical areas scan**
     The kernel memory, running processes, and disk boot sectors are scanned if you select this type.

- **Custom scan**

  In the **Specify a path** field, specify a path to the file that you want to scan. If you want to set several paths, click the **Add path** button, and then specify the path.

2. Click the **Scan** button.

The selected type of malware scan starts.

If the malware scan is completed successfully, an appropriate message is displayed on the screen, and the alert or incident is displayed in the alert table or incident table with the **Success** action status. Otherwise, an error message is displayed, and the alert or incident is displayed with the **Error** action status.

After the malware scan operation is finished, you can view the result.

YARA scan

A YARA scan is a search for malicious activity signatures on devices, by using YARA files, manually downloaded via OSMP Console.

To perform a YARA scan, you must have:

- Kaspersky Endpoint Security with a built-in agent installed on your devices
- At least one rule in the list of YARA rules

To perform a YARA scan:

1. Select the **YARA scan** response action.

   The **Run YARA scan** pane opens.

2. In the **Rules** drop-down list, select one or more YARA rules.

   You can enter the name of the rule or a sequence of characters from the name of the rule.

   The maximum number of rules you can specify is 200.

3. In the **Scan scope** do the following:

   a. Select the option and specify its mandatory parameters:

   - **RAM** to scan processes that are running at the time of the response action execution.

     This option is selected by default, and in the **Scan** group of settings the **All processes** option is selected.

     If you want to scan specific processes, in the **Scan** group of settings, select the **Specific processes** option, and then enter the name of one or more processes to be scanned. Otherwise, all processes will be scanned.

     The application does not scan processes with a low priority.

   - **Autorun points** scan autorun points obtained from the **Forensics** response action.

Then, specify the scan type:

- **Quick scan**. All autorun points are scanned, except COM objects.
- **Full scan**. All autorun points are scanned, as well as files involved with them.

List of supported autorun points

The following types of autorun points are supported:

- Logon
- Shell
- Office
- Browsers
- Scheduler
- Services
- Drivers
- Cryptography
- Debuggers
- COM
- SessionManager
- Network
- LSA
- Applications
- WMI
- GroupPolicy
- DriverLog
- BootLog
- Process
- OsUpdate
- Printer
- Unspecified

- OsLoader
- EfiLoader

- **All local disks** to scan files stored in all folders on local disks.

  Scanning all local disks can cause high load on the device.

- **Specified directories** to scan files that are located in a specified directory and all its nested folders.

  You have to specify the full path to the directory in the corresponding field, appending the mask `*` at the end to include subdirectories, for example, `C:\Users\Administrator\Desktop\*`.

b. If necessary, in the corresponding fields, specify the following parameters for the selected option:

- **Maximum scan duration (hours)**

  The default value is 23. After the specified time elapses, the scan is stopped even if some rules were not applied to scan the devices. The response action result contains data that is up-to-date at the moment when the scan was stopped.

- **Exclusions**

  Specify short names of processes or a mask of files that you want to exclude from scanning. If multiple processes with identical names are running on the device, the application excludes all such processes from scanning.

4. If necessary, configure the following settings:

- Specify the tenant.

  If you ran the response action from the list of YARA rules and selected rules that belong to different tenants, the **Tenant** field is empty. You have to specify the tenant manually for whose devices you want to perform a YARA scan.

- Change the scope of the response action.

  By default, in the **Scope** group of settings, the **Specific assets** option is selected, and the **Assets** drop-down list contains the names of the devices whose check box you previously selected. The YARA scan will be performed on these devices.

  You can either change the devices to be scanned or select the **All assets** option in the **Scope** group of settings to scan all the devices that belong to the alert or incident tenant and its child tenants.

  Only the assets on which you can perform the response action are available for selection.
  The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

- In the **Description** field, write a description or a comment to the response action.

5. Click the **Run** button.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response action in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

In the **Assets** column, you have to click the link with the number of assets for which the YARA scan was launched. In the window opens with the **Result** tab selected, you can view the scan status for every YARA rule. If the scan detects a threat, a corresponding alert is created. You can move to the alert from the response action result.

If the response action completed unsuccessfully, an **Error** message and information about the reason of the error are displayed.

IOC scan

An IOC scan searches for signs of targeted attacks, as well as infected and probably infected objects on devices, by using IOC files (indicators of compromise files) manually downloaded via OSMP Console.

To perform an IOC scan, you must have:

- Kaspersky Endpoint Security with a built-in agent installed on your devices
- At least one rule in the list of IOC rules

To perform an IOC scan:

1. Select the **IOC scan** response action.

   The **Run IOC scan** pane opens.

2. In the **Rules** drop-down list, select one or several IOC rules.

3. If necessary, configure the following settings:

   - Specify the tenant.

     If you ran the response action from the list of IOC rules and selected rules that belong to different tenants, the **Tenant** field is empty. You have to specify the tenant manually for whose devices you want to perform an IOC scan.

   - Change the scope of the response action.

     By default, in the **Scope** group of settings the **Specific assets** option is selected, and the **Assets** drop-down list contains the names of the devices whose check box you previously selected. This means the IOC scan will be performed on these devices.

     You can either change the devices to be scanned or select the **All assets** option in the **Scope** group of settings, to scan all the devices that belong to the alert or incident tenant and its child tenants.

     > Only the assets on which you can perform the response action are available for selection.
     > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

- In the **Maximum scan duration (hours)** field, specify the maximum scan duration.

  The default value is 23. After the specified time elapses, the scan is stopped even if some rules were not applied to scan the devices. The response action result contains data that is up-to-date at the moment when the scan was stopped.

- In the **Description** field, write a description or a comment to the response action.

4. Click the **Run** button.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response action in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

In the **Assets** column, you have to click the link with the number of assets for which the IOC scan was launched. In the window that opens with the **Result** tab selected, you can view the scan status for every IOC rule. If the scan detects a threat, a corresponding alert is created. You can move to the alert from the response action result.

If the response action completed unsuccessfully, the **Error** message and the information about the reason of the error are displayed.

Scanning files obtained from devices

After you obtain a file from the device or from quarantine directory, you can scan the file for threats.

> To perform the response action, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

You can run the file scan in one of the following ways:

- While performing the File response action.

  The **Scan file** option is enabled by default. For details, refer to the Getting a file article.

- From the **Quarantine** section of the main menu, by clicking the **Get & scan** button.

  > The option is available if Kaspersky Endpoint Security with a built-in agent is installed on your devices.

- From the **Files** section of the main menu or from the **Files** tab in the alert or incident details.

  The option is available for the files placed to **Files** section as a result of the File or Get file response actions and for the files that you manually uploaded.

- From the results of the following response actions:
  - File
  - Get file
  - Move to quarantine

Running scan from Files section or tab

To run the scan from Files section or tab:

1. In the main menu, do one of the following:
   - Go to **Operations** → **Repositories** → **Files**.
   - Go to **Monitoring**, and then select the **Alerts** or **Incidents** section. In the **ID** column, click the ID of the alert or incident that includes the files obtained as a result of the Get file and Obtaining files from quarantine response actions. In the alert or incident details that open, go to the **Files** tab.
2. In the table of files, do one of the following:
   - Select the check box next to the required file, and then in the toolbar click the **Scan** button.
   - Click the file name, and then in the file details pane that opens, click the **Scan** button.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Running scan from the results of other response actions

To run the scan from the results of other response actions:

1. In the main menu, do one of the following:
   - Go to **Monitoring** → **Response history**.
   - Go to **Monitoring**, and then select the **Alerts** or **Incidents** section. In the **ID** column, click the ID of the alert or incident from which you performed the Get file, Obtaining files from quarantine, or Moving files to quarantine response action. In the alert or incident details that open, move to the **History** tab, and then select the **Response history** tab to display the list of events.
2. In the **Action status** column, click the **Success** status link for the event for which you want to view the results of the required response action.
3. In the right pane that opens, click the link with the file name.

   The file details pane opens.
4. In the toolbar, click the **Scan** button.

   If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the file scan result

You can view the file scan results in one of the following ways:

- From the **Files** section in the main menu or from the **Files** tab in the alert or incident details
- From the **Response history** section in the main menu or from the **History** tab in the alert or incident details

  After you click the link in the **Action status** column, the window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message and the a password-protected ZIP archive with the scan result are displayed. You can download the archive to your computer. The password is the following: infected.

  Otherwise, the **Error** message and the information about the reason of the error are displayed.

  If you want to view the **Launch ID** and the JSON with the response action result, move to the **Debug data** tab.

Deleting a file

To perform the **Delete file** response action, you must have:

- One of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst
- Kaspersky Endpoint Security with a built-in agent installed on your devices

You can perform the response actions in one of the following ways:

- From the alert or incident details
- From the device details
- From a telemetry event

  You can open the event details from alert or incident details or from the **Threat hunting** section.
- From an investigation graph

  > This option is available if the investigation graph is built.

- From the table of assets
- From the **Response history** section

You can also configure the response action to run automatically when creating or editing a playbook.

Responding from the alert or incident details

To respond from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device with the file to be deleted.

   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device with the file to be deleted.

2. In the window that opens, go to the **Assets** tab.

   > You can also go to the **Observables** tab, select the check box next to the file that you want to delete, and then click **Delete file**.

3. Select the check box next to the device on which the file to be deleted is located.

   You can select several devices.

4. In the **Select response actions** drop-down list, click **Delete file**.

5. In the right pane that opens, do the following:

   - In the **Full path** field, specify the full path to the file that you want to delete.

   - In the **Assets** drop-down list, select the devices on which the file is located.

     If at step 2 you selected the **Assets** tab, the names of the devices whose check boxes you selected are specified in the **Assets** field by default.

     You can either change the devices or select the **All assets** option in the **Scope** group of settings to perform the response action for all the devices that belong to the alert or incident tenant and its child tenants.

     > Only the assets on which you can perform the response action are available for selection.
     > The devices are displayed as follows: devices belonging to the alert or incident are placed in the beginning of the **Assets** drop-down list, and then the devices belonging to the alert or incident tenant and child tenants are displayed alphabetically.

   - If necessary, configure the following settings in the corresponding fields:

     - Specify either SHA256 or MD5 hash of the file.

       > If at step 2 you selected the **Observables** tab, the hash is specified. You can change the value.

     - Write a description or a comment to the response action.

6. Click the **Delete** button.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the device details

To respond from the device details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device with the file to be deleted.

   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device with the file to be deleted.

2. In the window that opens, go to the **Assets** tab.

3. Click the name of the required device, and then in the drop-down list, select **View properties**.

4. Perform the same actions as described at steps 4–6 in *Responding from the alert or incident details*.

   The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the telemetry event

To respond from a telemetry event when opening it from the alert or incident details:

1. Do one of the following:

   - In the main menu, go to **Monitoring** → **Alerts**. In the **ID** column, click the ID of the alert that includes the device with the file to be deleted.

   - In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device with the file to be deleted.

2. In the window that opens, go to the **Details** tab, and do one of the following:

   - Click the name of the required event and select the device.

   - Click the **Find in Threat hunting** button to go to the **Threat hunting** section, and then select the required device.

3. Perform the same actions as described at steps 4–6 in *Responding from the alert or incident details*.

   The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

To respond from a telemetry event when opening it from the Threat hunting section:

1. In the main menu, go to **Monitoring** → **Threat hunting**.

   The table of events opens.

2. Click the name of the event whose details you want to open.

3. In the event details pane that opens, click the device name, and then from the drop-down list, select **Delete file**.

4. In the right pane that opens, configure the response action settings as described at steps 5–6 in *Responding from the alert or incident details*.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from an investigation graph

To scan a device from an investigation graph:

1. In the main menu, go to **Monitoring** → **Incidents**. In the **ID** column, click the ID of the incident that includes the device with the file to be deleted.

2. Click the **View on graph** button.

3. In the investigation graph that opens, click the device name to open the device details.

4. In the **Select response actions** drop-down list, click **Delete file**.

5. Perform the same actions as described at steps 5–6 in *Responding from the alert or incident details*.

If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the table of assets

To perform the response action from the table of assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The table of assets opens.

2. Select the check box next to the device for which you want to perform the response action.

   You can select several devices.

   > The archived devices cannot be selected.

3. In the **Select response actions** drop-down list, select **Delete file**.

4. In the right pane that opens, configure the response action settings as described at steps 5–6 in *Responding from the alert or incident details*.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding from the Response history section

To perform the response action from the Response history section:

1. In the main menu, go to **Monitoring** → **Response history**.

   The table that contains the response history for all alerts and incidents opens.

2. In the **Response** drop-down list, select **Delete file**.

3. In the right pane that opens, configure the response action settings as described at steps 5–6 in *Responding from the alert or incident details*.

The pane is closed. If the response action is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Viewing the response action result

You can view the result of the response actions in one of the following ways:

- From the **Response history** section in the main menu
- From the **History** tab in the alert or incident details

After you click the link in the **Action status** column, the window opens with the **Result** tab selected. If the response action completed successfully, the **Success** message is displayed. Otherwise, the **Error** message and the information about the reason of the error are displayed.

If you want to view the Launch ID and the JSON with the response action result, move to the **Debug data** tab.

> If the file has been blocked by another process, the result is displayed with the **Success** status but the file will be deleted only after the device is restarted. It is recommended to check whether the file is successfully deleted after the device is restarted.

Responding through Active Directory

You can integrate Kaspersky Next XDR Expert with the Active Directory services that are used in your organization. Active Directory is considered to be integrated with Kaspersky Next XDR Expert after the integration between Active Directory and KUMA is configured.

The process of configuring integration between Kaspersky Next XDR Expert and Active Directory consists of configuring connections to LDAP. You must configure connections to LDAP separately for each tenant.

As a result, if an alert or an incident occurs, you will be able to perform response actions in relation to the associated users of that tenant.

You can perform a response action through Active Directory in one of the following ways:

- From the alert or incident details
- From a telemetry event (if you open it from alert details)

- From an investigation graph

  This option is available if the investigation graph is built.

You can also configure a response action to run automatically when creating or editing a playbook.

To perform a response action through Active Directory, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

To perform a response action through Active Directory:

1. In the main menu, go to the **Monitoring** section, and then select the **Alerts** or **Incidents** section.

   If you want to respond from the telemetry event, select the **Alerts** section.

   If you respond from an investigation graph, select the **Incidents** section.

2. Click the ID of the required alert or incident.

3. In the window that opens, do one of the following:

   - If you want to respond through the alert or incident details, go to the **Assets** tab, and then click the name of the user.

   - If you want to respond through a telemetry event, go to the **Details** tab, and either click the name of the required event, and then select the user; or click the **Find in Threat hunting** button to go to the **Threat Hunting** section, and then select the required user.

   - If you want to respond through an investigation graph, click the **View on graph** button. In the investigation graph that opens, click the name of the user.

   The **Account details** window opens on the right side of the screen.

4. In the **Response through Active Directory** drop-down list, select an action that you want to perform:

   - **Lock account**

     If the user account is locked in response to the related alert or incident, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

   - **Reset password**

     If the user account password is reset in response to the related alert or incident, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

   - **Add user to security group**

     In the window that opens, in the mandatory field **Security group DN**, specify a full path to the security group to which you want to add the user. CN = Team, OU = Groups, OU = ExchangeObjects, DC = example, DC = com. Then click the **Add** button. Only one group can be specified within one operation.

     If the user is added to the security group in response to the related alert or incident, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

   - **Delete user from security group**

In the window that opens, in the mandatory field **Security group DN**, specify a full path to the security group from which you want to delete the user. For example, CN = Team, OU = Groups, OU = ExchangeObjects, DC = example, DC = com. Then click the **Delete** button. Only one group can be specified within one operation.

If the user is deleted from the security group in response to the related alert or incident, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

Responding by using third-party scripts

This section contains the response actions that you can perform via external systems by launching third-party scripts on remote client devices.

To do this, you must:

- Configure integration between Kaspersky Next XDR Expert and a script launch service.
- Create a playbook that will launch a script for responding.
- Download, and then use the scripts as described in the instructions provided in this section.

Responding through UserGate

UserGate includes features of unified threat management solutions and provides the following means of protection for your local network:

- Firewall
- Intrusion and attack protection
- Anti-virus traffic scanning
- Application control

UserGate UTM API 7 version is supported.

You can respond to alerts and incidents through UserGate if you previously configured integration between Kaspersky Next XDR Expert and script launch service, as well as created a playbook that will launch a script for responding. You can download the scripts by clicking this link.

Download scripts

The login and password to access UserGate are stored in the ug.py script. You can change the *endpoint*, *login*, and *password* values in this script.

Python 3.10 is required to run the scripts.

To perform a response action through UserGate, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst.

You can create playbooks that will perform the following response actions through UserGate:

- Block IP addresses, URL and domain names.

  UserGate will block IP addresses, URL and domain names as a result of the playbook launch.

- Log out the users.

  All users that are logged in to UserGate will be logged out as a result of the playbook launch.

To launch a script for responding through UserGate:

1. In the main menu, go to the **Monitoring** section, and then in **Alerts** or **Incidents** section, click the ID of the required alert or incident.

2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through UserGate.

3. Click the **Launch** button.

   The selected playbook launches the script for responding through UserGate.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through Ideco NGFW

Ideco NGFW is a solution that acts as a filter for the internet traffic in corporate and private networks. It allows you to block IP addresses and URLs detected by Kaspersky Next XDR Expert, if you previously configured integration between Kaspersky Next XDR Expert and the script launch service.

Ideco NGFW version 16.0 or later is supported.

The login and password to access Ideco NGFW are stored in the script for integration with Ideco NGFW. You can download the script by clicking the following link:

Download script

To use the script:

1. Install the script in one of the following ways:

   - Via pip, for example:

     ```
     pip install -r requirements.txt
     ```

   - From the WHL file, for example:

     ```
     pip install ./dist/kaspersky_xdr_ideco_integration-<version>-py3-none-any.whl
     ```

   - Offline installation.

     If you do not have internet access, you must install the script offline. In this case, do the following:

     a. Download the dependencies on a computer that has internet access, by running the following command:

     ```
     pip download -r requirements.txt
     ```

     b. Move the downloaded dependencies to the device on which you will run the script.

     c. Install the dependencies by using the command:

     ```
     pip install --no-index --find-links <folder_path_to_downloaded_dependencies> -r requirements.txt
     ```

2. Configure the script in one of the following ways:

   - Via the ENV file, for example:

     ```
     cp .env.sample .env
     nano .env
     ```

   - In the body of the script (ideco.py), edit the parameters in the following strings:

     ```
     BASE_URL: str = getenv("BASE_URL", "https://your-ip:your-port")
     LOGIN: str = getenv("LOGIN", "your-login")
     PASSWORD: str = getenv("PASSWORD", "your-password")
     IP_DENY_LIMIT: int = int(getenv("IP_DENY_LIMIT", 1000))
     ```

3. Add deny rules for the IP addresses detected by Kaspersky Next XDR Expert and for malicious URLs.

To add a firewall rule that will block IP addresses:

1. **Run the script by using the add_firewall_rule command**.

   The command has the following logic:

   1. Check if the IP addresses exist in the Ideco NGFW object list.

      If they exist, the current IP address is not added.

      If they do not exist, the current IP address is added.

   2. Check if the list of IP addresses named `XDR` exists.

      If the list exists, it is reused, and IP addresses are added to it.

      If it does not exist, a new list is created, and IP addresses are added to it.

3. Check if the firewall rule named `XDR` exists.

   If the firewall rule exists, it is reused, and the list of IP addresses from step 2 is added to it.

   If it does not exist, a new firewall rule is created, and the list of IP addresses from step 2 is added to it.

2. Specify the IP addresses that you want to block.

   By default, the maximum number of IP addresses is 1000. You can edit this value, as described at step 2 *Configure the script*.

   You must add valid IPv4 addresses, separated with commas and without spaces, for example:

   ```
   python ideco.py add_firewall_rule --ip_address "12.12.12.12, 13.13.13.13"
   ```

   The deny rule for the selected IPv4 addresses is added.

To add a filtering rule that will block malicious URLs:

1. **Run the script by using the add_content_filter_file command**.

   The command has the following logic:

   1. Check if a category named `XDR` exists.

      If it exists, the URLs are added to this category.

      If it does not exist, a new category is created, and then the URLs are added to it.

   2. Check if the content filtering rule named `XDR` exists.

      If the content filtering rule exists, the category from step 1 is added to it.

      If it does not exist, a new content filtering rule is created, and then the category from step 1 is added to it.

2. Specify the URLs that you want to block.

   The URLs must be separated with commas, and have http:// or https:// prefixes, for example:

   ```
   python ideco.py add_content_filter_rule --url "https://url_1.com, http://url_2.com.uk,
   http://qwerty.nl, http://zxc.xc"
   ```

   The deny rule for the specified URLs is added.

Responding through Ideco UTM

Ideco UTM is a solution providing the following means of protection for your corporate network:

- Firewall—Filtering network traffic, to protect the network from unauthorized access.
- Intrusion and attack protection—Identifying and blocking suspicious actions, to ensure system integrity.
- Anti-virus traffic scanning—Protecting against malware and malicious activities.

- Application control—Blocking or restricting execution of unauthorized applications.
- Web filtering—Restricting user access to websites that you consider unwanted.

Ideco UTM 15.7.35 version is supported.

You can respond to alerts and incidents by using Ideco UTM if you previously configured integration between Kaspersky Next XDR Expert and a script launch service, as well as created a playbook that will launch a script for responding. As a result of the playbook launch, Ideco UTM will block IP addresses, IP ranges, or URLs, depending on the action that you specify when creating a playbook.

To unblock the IP addresses, IP ranges, or URLs that have been blocked, you have to create and launch another playbook.

You can download the script by clicking this link:

Download script

The login and password to access Ideco UTM are stored in the env.sample configuration file. You have to copy the information from this file to a new ENV file that you create, and then specify the necessary parameters in the new file.

Python 3.10 is required to run the script.

To perform a response action through Ideco UTM, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

To launch a script for responding through Ideco UTM:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.
2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through Ideco UTM.
3. Click the **Launch** button.

   The selected playbook launches the script for responding through Ideco UTM.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through Redmine

Redmine is a web application for project management and issue tracking. It allows you to automate the scenario of working with issues in Redmine projects by using the script if you previously configured integration between Kaspersky Next XDR Expert and the script launch service.

Download the script by clicking this link:

Download script

To use the script:

1. Install the script in one of the following ways:

   - Via pip, for example:

     ```
     pip install -r requirements.txt
     ```

   - From the WHL file, for example:

     ```
     pip install ./dist/kaspersky_xdr_redmine_integration-1.0-py3-none-any.whl
     ```

   - Offline installation.

     If you do not have internet access, you have to install the script offline. In this case, do the following:

     a. Download the dependencies on a computer that has internet access, by using the following command:

     ```
     pip download -r requirements.txt
     ```

     b. Move the downloaded dependencies to the device on which you will run the script.

     c. Install the dependencies by using the following command:

     ```
     pip install --no-index --find-links <folder_path_to_downloaded_dependencies> -r
     requirements.txt
     ```

2. Configure the script in one of the following ways:

   - Via the ENV file, for example:

     ```
     cp .env.sample .env
     nano .env
     ```

   - In the body of the script (redmine.py), edit the parameters in the following strings:

     ```
     REDMINE_URL: str = getenv("REDMINE_URL", "http://<ip_or_hostname>")
     REDMINE_PORT: str = getenv("REDMINE_PORT", "8080")
     REDMINE_API_KEY: str = str(getenv("REDMINE_API_KEY", "<redmine_api_key>"))
     ```

You can use the script to work with issues in Redmine.

- If you want to create a new issue, run the following command:

  ```
  python redmine.py create_issue "project-identifier" "Issue subject" --description
  "Issue description text" --priority_id <id: int>
  ```

  Result:

  ```
  {"issue_id": 57}
  ```

- If you want to update an issue, run the following command:

  ```
  python redmine.py update_issue <issue_id: int> --subject "Subject text to be updated"
  --description "Description text to be updated" --priority_id <id: int>
  ```

  Result:

  ```
  {"status": "issue_updated"}
  ```

- If you want to get an issue, run the following command:

  ```
  python redmine.py get_issue <issue id: int>
  ```

  Result:

  ```
  {
  "subject": "86",
  "description": "18",
  "project_name": "Test project",
  "author_name": "Redmine Admin",
  "status_name": "backlog",
  "priority_name": "high",
  "start_date": "24.07.2023",
  "due_date": null,
  "created_on": "24.07.2023 10:56:15",
  "updated_on": "24.07.2023 17:18:38"
  }
  ```

Responding through Check Point NGFW

Check Point NGFW is a solution that acts as a filter for internet traffic in corporate networks. Integration with Check Point NGFW allows you to block IP addresses and URLs detected by Kaspersky Next XDR Expert.

Check Point NGFW includes features of unified threat management solutions and provides the following means of protection for corporate networks:

- Firewall—Filtering network traffic, to protect the network from unauthorized access.
- Intrusion and attack protection—Identifying and blocking suspicious actions, to ensure system integrity.
- Anti-virus traffic scanning—Protecting against malware and malicious activities.
- Application control—Blocking or restricting execution of unauthorized applications.
- Web filtering—Restricting user access to websites that you consider unwanted.

Check Point NGFW version R81.20 or later is supported.

You can respond to alerts and incidents through Check Point NGFW if you previously configured integration between Kaspersky Next XDR Expert and the script launch service, as well as created a playbook that will launch a script for responding. To unblock the IP addresses or URLs that have been blocked, you have to create and launch another playbook.

Python 3.10 is required to run the scripts.

To perform a response action through Check Point NGFW, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

You can download the scripts for responding by clicking the following link:

Download script

The login and password to access Check Point NGFW are stored in the file `.envSample`.

To use the script:

1. Install the script in one of the following ways:

   - Via pip, for example:

     ```
     pip install -r requirements.txt
     ```

   - Offline installation.

     If you do not have internet access, you must install the script offline. In this case, do the following:

     a. Download the dependencies on a computer that has internet access, by running the following command:

     ```
     pip download -r requirements.txt
     ```

     b. Move the downloaded dependencies to the device on which you will run the script.

     c. Install the dependencies by using the command:

     ```
     pip install --no-index --find-links <folder_path_to_downloaded_dependencies> -r requirements.txt
     ```

2. Configure the script in one of the following ways:

   - Via the ENV file, for example:

     ```
     cp .env.sample .env
     nano .env
     ```

   - In the body of the script (main.py), edit the parameters in the following strings:

     ```
     BASE_IP: str = getenv("BASE_IP", "your-ip")
     BASE_PORT: str = getenv("BASE_PORT", "your-port")
     LOGIN: str = getenv("LOGIN", "your-login")
     PASSWORD: str = getenv("PASSWORD", "your-password")
     ```

3. Add deny rules for the IP addresses detected by Kaspersky Next XDR Expert and for malicious URLs.

To add a firewall rule that will block IP addresses:

1. **Run the script by using the add_firewall_rule command**.

   The command has the following logic:

   1. Check if the IP addresses exist in the Check Point NGFW object list.

      If they exist, the current IP address is not added.

      If they do not exist, the current IP address is added.

   2. Check if the list of IP addresses named `XDR` exists.

If the list exists, it is reused, and IP addresses are added to it.

If it does not exist, a new list is created, and IP addresses are added to it.

3. Check if the firewall rule named `XDR` exists.

   If the firewall rule exists, it is reused, and the list of IP addresses from step 2 is added to it.

   If it does not exist, a new firewall rule is created, and the list of IP addresses from step 2 is added to it.

2. Specify the IP addresses that you want to block.

By default, the maximum number of IP addresses is 1000. You can edit this value, as described in the previous procedure at step 2 *Configure the script*.

You must add valid IPv4 addresses, separated with commas and without spaces, for example:

```
python main.py add_firewall_rule --ip_address "12.12.12.12, 13.13.13.13"
```

The deny rule for the selected IPv4 addresses is added.

To delete a firewall rule that blocks IP addresses:

1. **Run the script by using the delete_firewall_rule command**.

The command has the following logic:

1. Check if the IP addresses exist in the Check Point NGFW object list.

   If they exist, the current IP address is not added.

   If they do not exist, the current IP address is added.

2. Check if the list of IP addresses named `XDR` exists.

   If the list exists, it is reused, and IP addresses are added to it.

   If it does not exist, a new list is created, and IP addresses are added to it.

3. Check if the firewall rule named `XDR` exists.

   If the firewall rule exists, it is reused, and the list of IP addresses from step 2 is added to it.

   If it does not exist, a new firewall rule is created, and the list of IP addresses from step 2 is added to it.

2. Specify the IP addresses that you want to block.

By default, the maximum number of IP addresses is 1000. You can edit this value, as described in the previous procedure at step 2 *Configure the script*.

You must add valid IPv4 addresses, separated with commas and without spaces, for example:

```
python main.py delete_firewall_rule --ip_address "12.12.12.12, 13.13.13.13"
```

The deny rule for the selected IPv4 addresses is deleted.

To add a filtering rule that will block malicious URLs:

1. **Run the script by using the add_content_filter_file command**.

   The command has the following logic:

   1. Check if a category named `XDR` exists.

      If it exists, the URLs are added to this category.

      If it does not exist, a new category is created, and then the URLs are added to it.

   2. Check if the content filtering rule named `XDR` exists.

      If the content filtering rule exists, the category from step 1 is added to it.

      If it does not exist, a new content filtering rule is created, and then the category from step 1 is added to it.

2. Specify the URLs that you want to block.

   The URLs must be separated with commas, and have an http:// or https:// prefix, for example:

   ```
   python main.py add_content_filter_rule --url "https://url_1.com, http://url_2.com.uk,
   http://qwerty.nl, http://zxc.xc"
   ```

The deny rule for the specified URLs is added.

To delete a filtering rule that blocks malicious URLs:

1. **Run the script by using the delete_content_filter_file command**.

   The command has the following logic:

   1. Check if a category named `XDR` exists.

      If it exists, the URLs are added to this category.

      If it does not exist, a new category is created, and then the URLs are added to it.

   2. Check if the content filtering rule named `XDR` exists.

      If the content filtering rule exists, the category from step 1 is added to it.

      If it does not exist, a new content filtering rule is created, and then the category from step 1 is added to it.

2. Specify the URLs that you want to block.

   The URLs must be separated with commas, and have an http:// or https:// prefix, for example:

   ```
   python main.py delete_content_filter_rule --url "https://url_1.com,
   http://url_2.com.uk, http://qwerty.nl, http://zxc.xc"
   ```

The deny rule for the specified URLs is deleted.

To launch a script for responding through Check Point NGFW:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.

2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through Check Point NGFW.

3. Click the **Launch** button.

   The selected playbook launches the script for responding through Check Point NGFW.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through Sophos Firewall

Sophos Firewall is a solution providing the following means of protection for your corporate network:

- Firewall—Filtering network traffic, to protect the network from unauthorized access.
- Intrusion and attack protection—Identifying and blocking suspicious actions, to ensure system integrity.
- Anti-virus traffic scanning—Protecting against malware and malicious activities.
- Application control—Blocking or restricting execution of unauthorized applications.
- Web filtering—Restricting user access to websites that you consider unwanted.

Sophos Firewall 19.5 version is supported.

You can respond to alerts and incidents by using Sophos Firewall if you previously configured integration between Kaspersky Next XDR Expert and a script launch service, as well as created a playbook that will launch a script for responding. As a result of the playbook launch, Sophos Firewall will block IP addresses, IP ranges, or URLs, depending on the action that you specify when creating a playbook.

To unblock the IP addresses, IP ranges, or URLs that have been blocked, you have to create and launch another playbook.

You can download the script by clicking this link:

Download script

The login and password to access Sophos Firewall are stored in the env.sample configuration file. You have to copy the information from this file to a new ENV file that you create, and then specify the necessary parameters in the new file.

Python 3.10 is required to run the script.

To perform a response action through Sophos Firewall, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

To launch a script for responding through Sophos Firewall:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.
2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through Sophos Firewall.
3. Click the **Launch** button.

   The selected playbook launches the script for responding through Sophos Firewall.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through Continent 4

Continent 4 is a solution providing the following means of protection for your corporate network:

- Firewall—Filtering network traffic, to protect the network from unauthorized access.
- Intrusion and attack protection—Identifying and blocking suspicious actions, to ensure system integrity.
- VPN gateway—Creating secure tunnels for data transmission between your organization's networks.
- Access control—Managing user access to internal and external network resources, based on security rules and policies.
- Data encryption—Using cryptographic algorithms to protect the transmitted data.

Continent 4 version 4.1.7 is supported.

You can respond to alerts and incidents through Continent 4 if you previously configured integration between Kaspersky Next XDR Expert and a script launch service, as well as created a playbook that will launch a script for responding.

You can create playbooks that will perform the following response actions through Continent 4:

- Block IP addresses and URLs.

  Continent 4 will block IP addresses and URLs. To unblock the IP addresses or URLs that have been blocked, you have to create and launch another playbook.

- Blocking the Indicators of Compromise (hereinafter also referred to as IoCs).

  Continent 4 will block the observables that you specified in the playbook trigger.

You can download the script by clicking this link:

Download script

The login and password to access Continent 4 are stored in the env.sample configuration file. You have to copy the information from this file to a new ENV file that you create, and then specify the necessary parameters in the new file.

Python 3.10 is required to run the script.

To perform a response action through Continent 4, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

To launch a script for responding through Continent 4:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.

2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through Continent 4.

3. Click the **Launch** button.

   The selected playbook launches the script for responding through Continent 4.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through SKDPU NT

SKDPU NT is a solution for privileged accounts management.

SKDPU NT version 7.0.4 is supported.

You can respond to alerts and incidents through SKDPU NT if you previously configured integration between Kaspersky Next XDR Expert and a script launch service, as well as created a playbook that will launch a script for responding.

You can create playbooks that will perform the following response actions through SKDPU NT:

- Termination of the user session. The playbook will terminate all sessions of the user when suspicious activities are detected or security rules are broken.
- Blocking the user account. The playbook will block the user account and limit the user's access to the system.
- Revoking the user rights. The user will be removed from the privileged user group, and the user's rights will be revoked.

You can download the script by clicking this link:

Download script

The login and password to access SKDPU NT are stored in the env.sample configuration file. You have to copy the information from this file to a new ENV file that you create, and then specify the necessary parameters in the new file.

Python 3.10 is required to run the script.

To perform a response action through SKDPU NT, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

To launch a script for responding through SKDPU NT:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.
2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through SKDPU NT.
3. Click the **Launch** button.

   The selected playbook launches the script for responding through SKDPU NT.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through FortiGate

FortiGate is a solution providing the following means of protection for your corporate network:

- Firewall—Filters network traffic and prevents unauthorized access.
- Intrusion and attack protection—Identifies and blocks suspicious actions.
- Web filtering—Restricts user access to websites that you consider unwanted.

- Malware protection—Prevents malware infections.
- Email filtering—Blocks spam messages and suspicious emails.

FortiGate 7.6.0 version is supported.

You can respond to alerts and incidents by using FortiGate if you previously configured integration between Kaspersky Next XDR Expert and a script launch service, as well as created a playbook that will launch a script for responding. As a result of the playbook launch, FortiGate will block IP addresses, URLs, or domain names, depending on the action that you specify when creating a playbook.

To unblock IP addresses, URLs, or domain names that have been blocked, you have to create and launch another playbook.

You can download the script by clicking the following link:

Download script

The login, password, and API key to access FortiGate are stored in the env.sample configuration file. You have to copy the information from this file to a new ENV file that you create, and then specify the necessary parameters in the new file.

Python 3.10 is required to run the script.

To perform a response action through FortiGate, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

To launch a script for responding through FortiGate:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.
2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through FortiGate.
3. Click the **Launch** button.

   The selected playbook launches the script for responding through FortiGate.

   If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Responding through Palo Alto NGFW

Palo Alto NGFW is a solution providing the following means of protection for your corporate network:

- Firewall—Filtering network traffic, to protect the network from unauthorized access.
- Intrusion and attack protection—Identifying and blocking suspicious actions, to ensure system integrity.

- Malware protection—Preventing malware infections.
- Web filtering—Restricting user access to websites that you consider unwanted.

PAN-OS (the operating system that runs Palo Alto NGFW) version 11.1.6-h7 is supported.

You can respond to alerts and incidents through Palo Alto NGFW if you previously configured integration between Kaspersky Next XDR Expert and a script launch service, as well as created a playbook that will launch a script for responding. If you want to view the playbook examples, click this link.

As a result of the playbook launch, Palo Alto NGFW will block IP addresses or URLs, depending on the action that you specify when creating a playbook. To unblock IP addresses or URLs that have been blocked, you have to create and launch another playbook.

You can download the script by clicking this link:

Download script

You have to configure the script according to the instruction from the README file:

a. Rename the config.ini.example file in config.ini.

b. In the file, specify the address of the device on which Palo Alto NGFW is deployed, and the API key that you previously generated in the Palo Alto NGFW interface, or obtained by running the following command from the directory where the script is stored:

```
python main.py --get-api-key
```

If necessary, you can also change the default groups and policies where IP addresses or URLs are to be added.

To perform a response action through Palo Alto NGFW, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, or Tier 2 analyst.

To launch a script for responding through Palo Alto NGFW:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** sections, click the ID of the required alert or incident.
2. Click the **Select playbook** button, and then in the window that opens, select the playbook that you created for responding through Palo Alto NGFW.
3. Click the **Launch** button.

   The selected playbook launches the script for responding through Palo Alto NGFW.

   If the operation is successful, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The result of the playbook launch is available in the alert or incident details, on the **History** tab.

Viewing response history from alert or incident details

After you perform a response action, you can view the response history in one of the following ways:

- From the alert or incident details.
- From the Response history section.
- From a playbook details.
- From Quarantine section.

  When you select a check box next to the required file, and then click the **View file in response history** button, you move to the **Response history** section. You can view the response action as a result of which the file was moved to **Quarantine**.

To view the response action history from the alert or incident details:

1. In the main menu, go to the **Monitoring** section.
2. Open the **Alerts** or **Incidents** section, and then click the ID of the alert or incident for which the response action was performed.
3. In the window that opens, go to the **History** tab, and then select the **Response history** tab.

   The table of events is displayed and contains the following columns:

   - **Time**. The time when the event occurred.
   - **Launched by**. Name of the user who launched the response action.
   - **Events**. Description of the event.
   - **Response parameters**. Response action parameters that are specified in the response action.
   - **Asset**. Number of the assets for which the response action was launched. You can click the link with the number of the assets to view the asset details.
   - **Action status**. Execution status of the response action. The following values can be shown in this column:
     - **Awaiting approval**—Response action awaiting approval for launch.
     - **Launched**—Response action is in progress.
     - **Success**—Response action is completed without errors or warnings.
     - **Warning**—Response action is completed with warnings.
     - **Error**—Response action is completed with errors.
     - **Terminated**—Response action is completed because the user interrupted the execution.

- **Approval time expired**—Response action is completed because the approval time for the launch has expired.
- **Rejected**—Response action is completed because the user rejected the launch.

- **Playbook**. Name of the playbook in which the response action was launched. You can click the link to view the playbook details.
- **Response action**. Name of the response action that was performed.
- **Asset type.** Type of asset for which the response action was launched. Possible values: **Device** or **User**.
- **Asset tenant**. The tenant that is the owner of the asset for which the response action was launched.

4. If necessary, click the settings icon (⚙), and then select the columns to be displayed in the table.
5. If necessary, click the filter icon ( ), and then in the window that opens, specify and apply the filter criterion:
   - Add a new filter by clicking the **Add filter** button.
   - Edit a filter by selecting necessary values in the following fields:
     - **Property**
     - **Condition**
     - **Value**
   - Delete a filter.
   - Delete all filters by clicking the **Reset all** button.

Viewing the result of the malware scan

After the malware scan is finished, you can view its result in one of the following ways:

- From the alert or incident details
- From a response history
- From a playbook details

To view the result of the malware scan:

1. In the main menu, go to the **Monitoring** section, and then do one of the following:
   - If you want to view the result from alert or incident details, go to the **Alerts** or **Incidents** section, and then click the ID of the alert or incident for which malware scan was performed. In the window that opens, go to the **History** tab, and then select the **Response history** tab to display the list of events.
   - If you want to view the result from a response history, go to the **Response history** section.

- If you want to view the result of the malware scan from the playbooks table, go to the **Playbooks** section, and do one of the following:
  - Select the playbook for which the malware scan was performed. In the toolbar, click the **View history** button, and then select **View response history**.
  - Click the name of the playbook for which the malware scan was performed. In the window that opens, click the **View history** button, and then select **View response history**.

2. In the **Action status** column, click the status of the event for which you want to view the results of the malware scan.

   In the window that opens, a table of detections is displayed. In the **Administration Server** field, you can select the Administration Server for which a table of detections is displayed.

   The table contains the following columns:

   - **Device**. Device name or ID.
   - **Path**. Path to the file.
   - **Hash**. SHA256.
   - **Detection name**. Name of the detection that occurred on the device.
   - **Action status**. Threat processing result.
   - **User**. Account of the user who is associated with the detection.

Working with files obtained as a result of response actions

To view the **Files** section, you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator, SOC manager, Tier 1 analyst, Tier 2 analyst, Junior analyst, Approver, Auditor.

The **Files** section displays the files that you download manually and obtain as a result of performing the following response actions: File, Forensics, Registry key, NTFS metafiles, Process memory dump, Get file from quarantine.

While the space that you allocated when deploying Kaspersky Next XDR Expert is running out, the oldest files are deleted from the **Files** repository, and you are not notified about it. Files for which a threat was detected are not deleted. Such files are linked to the alerts created as a result of threat detection.

Viewing the table of files

To open the table of files, in the main menu, go to **Operations** → **Repositories** → **Files**. The table displays the files that are stored in the tenants to which you have access rights.

In the upper right part of the toolbar, you can group, filter, and search the data in the table as follows:

- Click the settings icon (⚙), and then select the columns to be displayed in the table.
- Click the filter icon ( ), and then specify and apply the filter criterion in the invoked menu.
- Click the search icon (🔍), in the **Search** field that displays, write what you want to find, and then press **ENTER**.

The table contains the following columns:

- **Name or full path**. If you downloaded the file manually, the file name is displayed in the column. If the file is obtained as a result of a response action, the full path to the file is displayed.
- **Size**. The size of the file.
- **MD5**. The MD5 hash of the file, if any.
- **SHA256**. The SHA256 hash of the file, if any.
- **Uploaded on**. The date when the file was added to the **Files** repository.
- **Updated on**. The date when the file was updated in the **Files** repository.
- **Tenant**. The tenant to which the file belongs.

  In the distributed solution mode, if a tenant was created on a child instance, the value in the **Tenant** filed is displayed with the child instance icon (🖧).
- **Source**. The name of the user who added the file to the **Files** repository or the name of the device from which the file was obtained as a result of a response action.
- **Adding method**. The way used to add the file to the **Files** repository.

  If the file was uploaded by a user, the **Uploaded by user** value is displayed in the column. If the file was obtained as a result of a response action, the response action name is displayed.
- **Scan status**. The result of the last performed scan of the file. The following values can be shown in this column:
  - **Not scanned**
  - **Scanning**
  - **Not detected**
  - **Detected**
  - **Error**
- **Description**. The column contains the information (if any) that you wrote in the **Description** field when configuring a response action parameters or uploading a file manually.

You can filter data in the **Adding method** and **Scan status** columns by clicking the column header and selecting the required method or status from the drop-down list.

Performing actions in the table of files

In the upper left part of the toolbar, you can perform the following actions:

- Click the **Refresh** button to update the data in the table.
- Upload a file from your computer to the **Files** repository by clicking the **Upload** button.

  In the right pane that opens, you must specify the tenant to which the file will belong in the **Tenant** field, click the **Upload files** link, and then select the file from your computer. If necessary, you can also add a comment or a description in the **Description** field, and enable the **Scan files after uploading** toggle switch if you want to scan the files for threats.

  After you click the **Upload** button, the pane is closed, and the uploaded file is displayed in the table. When uploading a large file or several files, you have to refresh the table so it displays the uploaded files.
- Save files to your computer. To do this, select the check box next to one or several files in the table, and then click the **Download** button.
- Scan the selected files for threats using the following technologies: YARA, Sandbox, and Anti-Malware Engine. To do this, select the check box next to one or several files in the table, and then click the **Scan** button.
- Delete files. To do this, select the check box next to one or several files in the table, and then click the **Delete** button.
- Go to the list of the alerts to which the files are related. To do this, select the check box next to one or several files in the table, and then click the **Find alerts** button. The alerts created as a result of threat detect and the alerts from whose details you ran the response action (if a file was placed to the **Files** repository as a result of a response action) are displayed.

The **Scan** option is not available and the file hash is not calculated for files obtained as a result if the following response actions: Forensics, Registry key, NTFS metafiles, Process memory dump.

Viewing the file details

You can open a file details by clicking the link in the **Name or full path** column in the table.

In the right pane that opens, the file metadata (for example, the file name or path, size, hash), the scan status, the scan result are displayed:

- If an archive was sent for scan, the file details display information for those files in the archive for which the threat was detected, as well as the hashes of such files and the status from each of the technology: Anti-Malware, Sandbox, YARA.
- If the file was not scanned, the corresponding information is displayed.
- If the scan is not available for the file, the file details display the metadata only.

If a file was manually uploaded and scanned for threats, in the file details, you can also view the report on scan by clicking the **Download report** link in the **Scan status** field.

In the file details, you can do the following:

- Perform the following actions by clicking the corresponding buttons in the toolbar: scan, download, delete files, find linked alerts, edit the file description, move to the response history.

  The **Show in response history** option is not available for the manually uploaded files. For a file obtained as a result of a response action, after you click **Show in response history**, you move to the **Response history** section where the table of events will be filtered by the file.

- Click the link with the MD5 or SHA256 hash, and then in the drop-down list select the action that you want to perform:

  - **Create prevention rule** for the file in which a threat was detected. After you select the option, the **Add prevention rule** pane opens where you can configure the settings of the **Add prevention rule** response action.

  - **Find in Kaspersky TIP** to check the file hash at Kaspersky Threat Intelligence Portal.

  - **View Sandbox report**. The option is available for files manually uploaded to the **Files** repository and scanned with the Sandbox technology.

  - **Find linked alerts**. The linked alerts are the ones created as a result of threat detect and the alerts from whose details you ran the response action (if a file was placed to the **Files** repository as a result of a response action).

  - **Find linked events** to move to the **Threat hunting** section where the events table is filtered by the hash of your file.

Playbooks

Open Single Management Platform uses playbooks that allow you to automate workflows and reduce the time it takes to process alerts and incidents.

Playbooks respond to alerts or incidents according to the specified *algorithm*. Playbook launches an algorithm that includes a sequence of response actions that help analyze and handle alerts or incidents. You can launch the playbook manually or configure the automatic launch of the playbook you need.

The automatic launch of playbooks is performed according to the *trigger* that you configure when creating a playbook. A trigger defines the conditions that an alert or incident must meet to launch this playbook automatically.

One playbook scope is limited to only alerts or only incidents.

Playbooks cannot be launched automatically for child incidents. For child incidents, you can only launch the playbook manually. Before launching, you need to select the assets and observables for which the playbook must be launched. Note that the playbook can only be launched for objects that belong to the alert or incident tenant.

Note that the playbook can only belong to one tenant and it is automatically inherited by all child tenants of the parent tenant, including child tenants that will be added after the playbook is created. You can disable playbook inheritance by child tenants when creating or editing a playbook.

In Open Single Management Platform, there are two types of playbooks:

- Predefined playbooks

  Predefined playbooks are created by Kaspersky experts. These playbooks are marked with the [KL] prefix in the name and cannot be edited or deleted.

  By default, predefined playbooks operate in the **Training** operation mode. For more information, refer to the Predefined playbooks section.

- Custom playbooks

  You can create and configure playbooks yourself. When creating a custom playbook, you need to specify a playbook scope (alert or incident), a trigger for launching the playbook automatically, and an algorithm for responding to threats. For details about creating a playbook, see Creating playbooks.

Operation modes

You can configure both automatic and manual launch of playbooks. The way to launch the playbook depends on the selected operation mode.

These are the following types of operation modes:

- **Auto**. A playbook in this operation mode automatically launches when corresponding alerts or incidents are detected.

- **Training**. When corresponding alerts or incidents are detected, a playbook in this operation mode requests the user's approval to launch.

- **Manual**. A playbook in this operation mode can only be launched manually.

User roles

You grant user rights to manage playbooks by assigning user roles to the users.

The table below shows access rights for managing playbooks and performing the user actions.

User role	User right				
	Read	Write	Delete	Execute	Response confirmation
Main administrator	✓	✓	✓	✓	✓
SOC administrator	✓	✓	✓	—	—
Junior analyst	✓	—	—	—	—
Tier 1 analyst	✓	—	—	✓	—
Tier 2 analyst	✓	✓	✓	✓	—
SOC manager	✓	—	—	—	—
Approver	✓	—	—	—	✓
Auditor	✓	—	—	—	—
Tenant administrator	✓	✓	✓	✓	✓

Viewing the playbooks table

The playbooks table is displayed in the **Monitoring** → **Playbooks** section. By default, the table displays the playbooks related to all of the tenants to which you have access rights.

> The playbooks table displays all existing playbooks, except for playbooks in the **Deleted** status.

To configure the playbooks table, do any of the following:

- Apply tenant filter:

  a. Click the link next to the **Tenant filter** setting.

  b. The tenant filter opens.

  c. Select the check boxes next to the required tenants.

- Filter the data of the playbooks table:

  a. Click the filter ( ) icon.

  b. On the **Filters** tab, specify and apply the filter criterion in the invoked menu.

- If you want to hide or display a column, click the settings ( ) icon, and then select the necessary column.

The playbooks table is configured and displays the data you need.

The playbooks table contains the following information:

- **Name**. Name of the custom or predefined playbooks.

  > The predefined playbooks are marked with the [KL] prefix in the name and cannot be edited or deleted.

- **Status**. Playbook status. Possible values:

  - **Active**. Playbook is created and ready to use (all necessary web plug-ins are installed).

  - **Draft**. Playbook is created and saved as a draft. The playbook cannot be launched but is available for viewing, editing, customizing, and deleting in the tenant to which the playbook belongs.

  - **Deleted**. Playbook was deleted. By default, playbooks in this status are not displayed in the table. If you want to view them, click the **Status** column, and then in the drop-down list, select **Deleted**.

    > Only active and unavailable playbooks have the **Deleted** status after deletion. Playbooks in the **Draft** status are deleted permanently and not displayed in the playbooks table.

  - **Unavailable**. Playbook algorithm uses an unavailable web plug-in or contains response actions that are not allowed by your license.

- **Operation mode**. Playbook operation mode that defines the way to launch the playbook. For more details on operation modes, see the Playbooks section.
- **Tags**. Tags that are assigned to a playbook. You can filter playbooks by using the assigned tags.
- **Actions**. Actions that are launched within playbooks.
- **Launches**. Total number of playbook launches.
- **Modified**. Date and time of the last edit of the playbook.
- **Created**. Date and time the playbook was created.
- **Parent tenant**. Name of the tenant to which the playbook belongs.
- **Description.** Playbook description or a comment. By default, this column is hidden.
- **Scope**. Playbook scope. Possible values: **Alert** or **Incident**.
- **Created by**. Name of the user who created the playbook. By default, this column is hidden.
- **Updated by**. Name of the user who edited the playbook. By default, this column is hidden.

Creating playbooks

You can create a playbook to automate threat analysis and threat response.

> To create a playbook, you must have one of the following roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

Kaspersky Next XDR Expert also allows you to create a new playbook that will meet your needs, based on an existing one. For more details, refer to Customizing playbooks.

To create a new playbook:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. Click the **Create** button.

   The **Create playbook** window opens with the visual editor work area on the left and the **Playbook settings** pane on the right. The visual editor allows you to build and visualize the playbook execution flow, and the relationship between the steps involved in playbook creation.

   > Based on the actions that you perform in the visual editor, the DSL schema of the playbook algorithm is generated. The changes you make in the visual editor working area are displayed in code, and vice versa. If necessary, you can open the code editor by clicking the **Show code** button in the toolbar.

3. In the **Playbook settings** pane, specify the following settings:

- In the **Tenant and inheritance** field, select a parent tenant and child tenants for which the playbook should be launched.

  All child tenants of the selected parent tenant will automatically inherit this playbook. To disable playbook inheritance, clear the check box next to a required child tenant. Playbook inheritance will be disabled for this tenant as well as for its subtenants, if any.
  If you select a child tenant, its subtenants will be selected automatically.

- In the **Name** field, enter the playbook name.

  Note that the playbook name must be unique within a tenant and cannot be more than 255 characters long.

- In the **Scope** list, select one of the following options:

  - **Alert**. The playbook will be launched only for alerts. This option is selected by default.

  - **Incident**. The playbook will be launched only for incidents.

- In the **Operation mode** list, select one of the following options:

  - **Auto**. A playbook in this operation mode automatically launches when corresponding alerts or incidents are detected.

  - **Training**. When corresponding alerts or incidents are detected, a playbook in this operation mode requests the user's approval to launch.

  - **Manual**. A playbook in this operation mode can only be launched manually.

- If necessary, in the **Tags** field, specify up to 30 tags. You can filter playbooks by using the assigned tags.

  Note that the maximum tag length is 50 characters.

- If necessary, in the **Description** field, enter a playbook description or a comment.

- In the **When launching several playbook instances at the same time** drop-down list, choose an action to perform if two or more playbook instances are launching at the same time:

  - **Add new playbook instances to the queue**. A new playbook instance will be launched after the current one is completed. By default, this action is selected.

  - **Terminate current execution and launch a new instance**. The execution of the current playbook instance will be terminated. After that, a new playbook instance is launched.

  - **Do not launch new playbook instances**. A new playbook instance will not be launched. The execution of the current playbook instance will continue.

  The **When launching several playbook instances at the same time** list is displayed only if the **Auto** operation mode is selected.

- By default, the playbook will only be launched for new alerts or incidents that match the trigger.

  If you want to launch a new playbook for existing alerts or incidents that match the trigger, select the **Launch the playbook for all matching alerts or incidents. Note that the system may be overloaded** check box.

- If necessary, in the **Additional settings** section, specify the following playbook parameters:

  - **Playbook timeout**—The maximum execution time of the playbook. By default, the value is 24h. The maximum value is 48h.

    If you want to create policies to be further applied in response actions, enable the **Use timeout policies for the response action** toggle switch, and then add the policies.

    > After you disable the toggle switch, the added policies are deleted.

  - **Input**—A jq expression to modify incoming data of the playbook before execution.

  - **Output**—A jq expression to modify the output data of the playbook after execution.

  - **Change method**—How to change the output data. Select whether to merge current and new data or overwrite current data with new data by selecting the **Merge** or **Overwrite** option.

  The other additional settings are predefined and cannot be changed:

  - **Playbook version**—Version of the playbook. May be required to identify the playbook in telemetry more accurately.

  - **DSL version**—DSL schema version.

  - **Action version**—Response actions schema version with the description of the response function input parameters.

4. In the visual editor work area, configure the playbook trigger and algorithm. Click the **Trigger** step, and then in the trigger settings pane that opens, specify the condition for the automatic launch of the playbook.

   > If you select the **Manual** operation mode, trigger settings are unavailable.

   To describe the trigger condition, use jq expressions. For more information about jq expressions, refer to jq Manual.

   Depending on the option you select in the **Scope** list when creating or editing a playbook, alert data model or incident data model is used.

   For example, to filter alerts or incidents by critical severity, specify the following expression:

   ```
   .Severity == "critical"
   ```

   You can also specify complex expressions to filter alerts or incidents.

   For example, to filter critical alerts or incidents by rule name, specify the following expression:

   ```
   (.Severity == "critical") and (.Rules[] |.Name | contains("Rule_1"))
   ```

   where `Rules[] |.Name` defines the name of the triggered rule.

Validation of jq expressions is configured. If you specify an incorrect expression when configuring the playbook trigger, the error is marked in red. If you want to view the details, hover the mouse cursor over the error.

5. If you want to view alerts or incidents that match the playbook trigger, in the **Trigger matching** section, click the **Find** button. By default, the last 10 modified alerts and incidents are displayed. If you want to change the value, open the drop-down list, and then select the required value.

   You can also request a full list of alerts or incidents. To do this, in the **Trigger** section, enter `true`, and then click the **Find** button.

   The full list of alerts or incidents is displayed.

6. Specify the playbook algorithm steps.

   You can copy an algorithm from another playbook. To do this, in the toolbar, click the **Copy algorithm** button, then in the window that opens, select a playbook from which to copy the algorithm, and then click **OK**.

   Validation of jq expressions and JSON syntax is configured. If you specify an incorrect expression when configuring the playbook algorithm, the error is marked in red. If you want to view the details, hover the mouse cursor over the error.

   To add an algorithm step, do the following:

   a. Hover the mouse cursor over the step from which a relation with the new step is to be built.

      If you are adding a step for the first time, you can hover the mouse cursor over the **Trigger** step only.

   b. Click and hold the white dot sign (○) that displays, and then start drawing an arrow.

   c. Release the mouse cursor, and then in the menu that opens, select the step.

      The selected step is displayed in the visual editor.

   d. Click the step to configure its settings. For more details, refer to Configuring playbook execution steps in visual editor.

   Alternately, you can right-click any part of the visual editor work area to invoke the menu, and then select the step you want to add. In this case, after adding the step, you have to build the relationships by drawing arrows between the required steps. If no relationships are built, you can only save the playbook as a draft.

   If algorithm steps are not added, you can save the playbook only as a draft.

7. Save the playbook in one of the following ways:

- By clicking the **Publish** button. The saved playbook is ready to use and displayed in the playbooks table, with the **Active** status.

- By clicking the **Create as draft** button. This allows you to create a playbook, by specifying only its name and the tenant to which it belongs. The playbook is displayed in the list of playbooks with the **Draft** status. You can open it and continue editing.

  When saving a playbook as a draft, validation is performed, and the errors are highlighted but ignored. You can save the playbook only as a draft when the relationships between the execution steps and the trigger are not built, the mandatory settings are not configured, or you specified invalid parameters.

A new playbook is created and displayed in the list of playbooks.

Configuring playbook execution steps in visual editor

After you add a playbook execution step when creating or editing a playbook, you have to configure its settings. To do this, click the block with the step, and then in the right pane, specify the required parameters by using jq expressions.

In visual editor, you can add and configure the following types of playbook execution steps:

- **Action**

  - In the **Action type** list, select the type of response action. The other settings vary based on the selected type.

    The types of response actions can be conventionally divided **into groups.**

    KATA/KEDR response actions:

    - **Add prevention rule**

      Specify the assets to which the action should be applied, and the SHA256 or MD5 hashes of the files that you want to block.

    - **Delete prevention rule**

      Specify the assets on which the prevention rule was applied, and the file hashes to be deleted.

    - **Reset prevention rules**

      Specify the assets on which the prevention rules are to be reset.

    - **Run program**

      Specify the assets and the executable file parameters: the full path to the file in the **Full path** field, and the parameters with which you want to start the file in the **Command line parameters** field. In the **Working directory** file, you can specify the path to the directory in which you want to save the results of the command or file execution.

    - **Enable network isolation**

      Specify the assets for which you want to enable isolation, and the period of isolation. If you want to add an exception from the network isolation rule, click the **Add exclusion** button, and then specify the required parameters.

- **Disable network isolation**

  Specify the assets on which the network isolation was applied.

Active Directory response actions:

- **Block LDAP account**

  Specify the users whose account you want to block.

- **Reset LDAP password**

  Specify the users whose password is to be reset.

- **Add to Active Directory group**

  Specify the users and the group to which you want to add the users.

- **Delete from Active Directory group**

  Specify the users and the group from which you want to remove the users.

Kaspersky NGFW response actions:

- **Add IP to blocklist on network devices**

  Specify the IP addresses that you want to block.

KES for Windows response actions:

- **Run malware scan**

  Specify the assets and the scan type:

  - **Critical areas scan**.
  - **Custom scan**, and then specify a path to the file that you want to scan.
  - **Full scan**, and if necessary, include network devices into the scan by enabling the **Scan network drives** toggle switch.

  If you want the next response action to be launched only after the malware scan, enable the **Wait for the response action to complete** toggle switch.

- **Update databases**

  Specify the devices on which you want to update databases. If you want the next response action to be launched only after the databases update, enable the **Wait for the response action to complete** toggle switch.

- **Terminate process**

  Specify the devices on which you want to terminate a process. Then, select one of the parameters and specify the jq expression with the required information: **Hash and path** to the process file, **Full path** to the file, or the **PID** (process ID).

  When Kaspersky Endpoint Security with a built-in agent operates in EDR Expert mode, terminating processes by system PID is not supported.

- **Move to quarantine**

  Specify the devices on which the file to be moved is located. Then, select one of the parameters and specify the jq expression with the required information: **Full path** to the file you want to quarantine, or **Hash and path** to the file.

KASAP response action:

- **Assign KASAP group**

  Specify the users to be moved to the KASAP learning group, and then specify the group ID.

Kaspersky Security Center response action:

- **Move to group**

  Specify the devices and path to the group to which you want to move the devices.

KICS for Networks response action:

- **Change authorization status**

  Specify the devices and their status: *authorized* or *unauthorized*.

Custom third-party response action:

- **Run custom script**

  In the corresponding fields, specify the command to run, jq expressions with command-line parameters to be applied to the command, and the path to the working directory. If a script requires additional data from the standard input, specify this data in the **Standard input** field.

Kaspersky Threat Intelligence Portal response action:

- **Enrich data**

  Select the number of objects to enrich: all objects or the top 100 objects from the source. Specify a jq expression with an array of observables to enrich. You can enrich the following types of observables: MD5, SHA256, URL, domain, or IP address.

Editing alerts or incidents by using a playbook:

- **Assign to user or group**

  Select the users or groups to whom you want to assign an alert or an incident.

- **Change external reference attribute**

  Specify a jq expression with the alert or incident external reference attribute, and then select whether to overwrite the current output data or merge the current and new data.

- **Change additional data**

  Select whether to overwrite the current output data, or merge the current and new data.

- **Change status**

  Select the status of the alert or incident: **New**, **In progress**, or **Closed**. If the playbook scope is incidents, you also have to select the incident type.

- **Change type**

  Select the type for the incident.

This action type is available only if you selected **Incident** in the **Scope** drop-down list when creating or editing a playbook.

- **Change description**

  Specify a jq expression with the incident description, and then select whether to overwrite the current output data, or merge the new and the current data.

  This action type is available only if you selected **Incident** in the **Scope** drop-down list when creating or editing a playbook.

- **Change priority**

  Select the priority for the incident: **Critical**, **High**, **Medium**, or **Low**.

  This action type is available only if you selected **Incident** in the **Scope** drop-down list when creating or editing a playbook.

- **Add a comment**

  Specify a jq expression with a comment to the alert or incident.

- If necessary, in the **Additional settings** section, click the chevron icon (>) and **specify additional settings**.

  - Click the **Terminate** or **Continue** button to select whether to terminate or continue the execution of the playbook if an error occurs.
  - Configure the response action timeout. You can either select the **Policy** option, and then specify the required timeout policy; or select the **Custom** option, and then specify the timeout in hours.
  - Modify the playbook outgoing data by enabling the **Output data** toggle switch. In the **Output** field, specify the jq expressions to replace the data returned by the response action, and then select the **Merge** or **Overwrite** replace method.
  - Manually approve the response function call by enabling the **Manual approve** toggle switch, and then specifying the approval timeout in minutes. The default value is 60 minutes.

    If you want to enable email notification, select the **Notify by email** check box, and then specify the delay before sending the email notification. The minimum value is 5 minutes, the maximum value is 30 minutes. Also, make sure that the SMTP server connection is configured and the email address is specified in the user's properties.

- Loop

  After you select this type of step, the **Loop** and **End loop** blocks are added to the visual editor.

  - Click the **Loop** block, specify a jq expression to form an array of incoming data, and then select the **Parallel** or **Sequence** mode for processing the elements.
  - If necessary, in the **Additional settings** section, click the chevron icon (>) and **specify additional settings.**

    - Click the **Terminate** or **Continue** button to select whether to terminate or continue the execution of the playbook if an error occurs.
    - In the **Maximum batch size** field, specify the number of elements that a playbook can take as an input.

- Set the rules for aggregating the results of the playbook execution over the elements of the incoming array. After you enable the **Aggregate results** toggle switch, in the **Expression** field, specify the jq expression to use the outgoing data as an input data for the next execution steps.
- Modify the outgoing data by specifying the jq expression in the **Output** field.
- Click the **Merge** or **Overwrite** button to select whether to overwrite the current output data, or merge the new and the current data.

- Parallel

  After you select this type of step, the **Parallel** and **End parallel** blocks are added to the visual editor.

  - Click the **Parallel** block, specify a jq expression to form an array of incoming data, and then enter the names of the branches.

    You can add new branches by clicking the **Add branch** button.

  - If necessary, in the **Additional settings** section, click the chevron icon (>) and **specify additional settings**.

    - Click the **Terminate** or **Continue** button to select whether to terminate or continue the execution of the playbook if an error occurs.
    - Set the rules for aggregating the results of the playbook execution over the elements of the incoming array. After you enable the **Aggregate results** toggle switch, in the **Expression** field, specify the jq expression to use the outgoing data for the next execution steps.
    - Modify the outgoing data by specifying the jq expression in the **Output** field.
    - Click the **Merge** or **Overwrite** button to select whether to overwrite the current output data, or merge the new and the current data.

- Decision

  After you select this type of step, the **Decision** and **End decision** blocks are added to the visual editor. Click the **Decision** block, specify a jq expression with a condition, and the condition name.

  If necessary, add new conditions by clicking the **Add condition** button.

- Update data

  Specify the jq expressions in the **Output** field, and then select whether to overwrite the current output data or merge the new and the current data.

You can perform the following actions with playbook execution steps:

- Editing by clicking the execution step block, and then in the properties window that opens, changing the required parameters.

  For parameters that imply entering jq expressions, syntax is highlighted and suggestions are displayed when you start typing.

- Moving through the visual editor work area by clicking and dragging the execution step block.
- Deleting by clicking the execution step block, and then pressing `Backspace` or `Delete`.

In the toolbar, you can perform the following actions:

- Opening the **Playbook settings** pane by clicking the **Settings** button.
- Opening the code editor with the DSL schema by clicking the **Show code** button.
  You can expand the code panel by clicking the (⌃) icon, and close the panel by clicking the (×) icon.
- Copying an algorithm from another playbook by clicking the **Copy algorithm** button.
- Opening the playbook response history by clicking the **View response history** button.
- Emulate the playbook launch without actually performing response actions, by clicking the **Launch in test mode** button.
- Undoing any action in visual editor by clicking the **Undo** button.
- Reverting a previously undone action by clicking the **Redo** button.
- Automatically aligning the elements by clicking the **Arrange** button.

Editing playbooks

To edit a playbook, you must have one of the following roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

For predefined playbooks, you can only change the playbook mode and launching rule. You can also view alerts or incidents that match the predefined playbook.

You can edit a playbook only if you have permission to change the tenant to which the playbook belongs.

To edit a playbook:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. Click the name of the playbook that you want to edit.
3. In the window that opens, edit the playbook's properties. For more details on the playbook properties that you can edit, see Creating playbooks.

4. If you changed the operation mode to **Auto**, in the **When launching several playbook instances at the same time** drop-down list, select one of the following actions:

   - **Add new playbook instances to the queue**. A new playbook instance will be launched after the current one is completed. By default, this action is selected.

   - **Terminate current execution and launch a new instance**. The execution of the current playbook instance will be terminated. After that, a new playbook instance is launched.

   - **Do not launch new playbook instances**. A new playbook instance will not be launched. The execution of the current playbook instance will continue.

5. If you changed the operation mode to **Manual**, in the **Action on launched playbook instances** drop-down list, select an action to apply to launching playbook instances:

   - **Terminate instances that are in progress or awaiting approval**.

   - **Terminate only the instances that are awaiting approval**.

   - **Execute all instances that are in progress or awaiting approval**.

6. Save the changes by clicking the **Save** button.

   You can also click the **Save with comment** button if you want to add a comment to the playbook before saving.

The playbook's properties are modified and saved.

Customizing playbooks

You can customize any playbook to your needs.

To customize playbooks:

1. In the main menu, go to **Monitoring** → **Playbooks**.

2. Open the playbook for editing by selecting the playbook from the list, and then clicking the **Duplicate and edit** button.

3. In the window that opens, configure the playbook's properties according to your needs.

   For more details on the playbook's properties that you can edit, refer to Creating playbooks.

   If you want to customize the playbook algorithm parameters, refer to Playbook algorithm.

   > The name of the customized playbook must be unique.

4. Click the **Save** button.

The customized playbook is modified and saved.

Viewing playbook properties

Playbooks allow you to automate workflows and reduce the time it takes to process alerts and incidents.

> To view a playbook, you must have one of the following roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, SOC manager, Approver, Auditor, Tenant administrator.

To view a playbook's properties:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. In the list of playbooks, click the name of the playbook that you want to view.

   The window that opens contains information about the playbook.

In the **Playbook settings** pane on the right, the following general information is displayed:

- **Tenant and inheritance**. Name of the tenant to which the playbook belongs.
- **Name**. Name of the playbook.
- **Scope**. Playbook scope. Possible values: **Alert** or **Incident**.
- **Operation mode**. Playbook operation mode. Possible values: **Auto**, **Training**, **Manual**.

  If the playbook operation mode is set to **Auto**, the **When launching several playbook instances at the same time** filed is displayed. Possible values: **Do not launch new playbook instances**, **Terminate current execution and launch a new instance**, **Add new playbook instances to the queue**.
- **Tags**. Tags assigned to the playbook.
- **Description**. Playbook description.
- **Additional settings**. Optional settings specified when creating the playbook.

In the visual editor work area on the left, the playbook trigger and algorithm (playbook execution steps) are displayed.

> You can edit the playbook's properties if you have one of the following roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

You can view the playbook response history by clicking the **View response history** button in the toolbar. The list of all playbooks or response actions launched within the playbook opens. You can terminate any launched playbooks or response actions by clicking the **Terminate** button.

If necessary, you can group and filter the data in the table as follows:

- Click the settings icon (⚙), and then select the columns to be displayed in the table.
- Click the filter icon ( ), and then specify and apply the filter criterion in the invoked menu.

  When you apply the filter criterion for the **Status** column, the table displays the manually launched responses whose status contains the selected value and the playbooks that include response actions whose status contains the selected value. It means that only the response actions of the playbook that meet the filter criterion will be displayed.

The filtered table of playbooks is displayed.

The table contains the following columns:

- **Actions**. Response action name.
- **Response parameters**. Response action parameters that are specified in the playbook algorithm.
- **Start**. Date and time the playbook or response action was launched.
- **End**. Date and time the playbook or response action was completed.
- **Alert or incident ID**. ID that contains a link to the alert or incident details.
- **Launched by**. Name of the user who launched the playbook or response action.
- **Approver**. Name of the user who approved the launch of the playbook or response action.

  By default, this column is hidden. To display the column, click the settings icon (⚙), and then select the **Approver** column.
- **Approval time**. Date and time when the user confirmed or rejected the launch of the playbook or response action.

  By default, this column is hidden. To display the column, click the settings icon (⚙), and then select the **Approval time** column.
- **Status**. Execution status of the playbook. The following values can be shown in this column:
  - **Awaiting approval**—Playbook awaiting approval for launch.
  - **In progress**—Playbook is in progress.
  - **Success**—Playbook is completed without errors or warnings.
  - **Warning**—Playbook is completed with warnings.
  - **Error**—Playbook is completed with errors.
  - **Terminated**—Playbook is completed because the user interrupted the execution.
  - **Approval time expired**—Playbook is completed because the approval time for the launch has expired.
  - **Rejected**—Playbook is completed because the user rejected the launch.

You can click the **Status** value to open the window with the result of the playbook launch. The Launch ID can be used by Technical Support to download the data necessary for analysis in case of the execution errors. By default, the data is stored for 72 hours.
If the status is **In progress**, you can view the Launch ID by hovering the mouse cursor over the icon next to the status.

- **Assets**. Number of the assets for which the playbook or response action is launched. You can click the link with the number of the assets to view the asset details. The field is empty, if the playbook or response action does not involve assets.

You can also view response history from the **Response history** section or from alert or incident details.

Terminating playbooks

You can forcibly terminate the launched playbook. In this case, the uncompleted response actions will be terminated. The completed response actions will not be canceled after the termination of the playbook.

To terminate a playbook, you must have one of the following roles: Main administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

To terminate a playbook:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. In the list of playbooks, do one of the following:
   - Select the required playbook, and then in the toolbar click the **View history** button.
   - Click the name of the required playbook, and then, in the window that opens, click the **View history** button.
3. Select **View response history**.
   The pane with the list of the launched playbook instances opens.
4. Select one or several instances that you want to terminate, and then click the **Terminate** button, and confirm your selection.

The playbook is terminated.

Deleting playbooks

Predefined playbooks cannot be deleted.

To delete a custom playbook, you must have one of the following roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

To delete a custom playbook:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. Select the playbook from the list, and then click the **Delete** button.
3. In the confirmation dialog box, click **Delete**.

> The playbook cannot be deleted if there are launched playbook instances. In this case, terminate all launched instances before deleting the playbook.

If you deleted a playbook in the **Active** or **Unavailable** status, the playbook is displayed in the playbooks table with the **Deleted** status and is only available for viewing and copying.

> Playbooks in the **Deleted** status are deleted permanently.

Launching playbooks in test mode

You can launch playbooks in test mode to debug them or validate the result. During a test run, the playbook processes selected alerts and incidents but does not perform any actions on assets.

Launching in test mode is available for playbooks with any status.

To launch a playbook in test mode, you must have one of the following roles: Main administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

To launch a playbook in test mode:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. Do one of the following:
   - Click the name of the playbook that you want to edit. In the **Playbook details** window that opens, click the **Launch in test mode**.
   - Select the playbook from the list, and then click the **Launch in test mode** button.

     Depending on the playbook's scope, either a **Select incident** or a **Select alert** is invoked.
3. Select the incident or the alert in the invoked panel.

   If the playbook requires selecting observables and assets, enable the **Select target objects before launching the playbook** option.
4. Click **Launch**.

   If you enabled the **Select target objects before launching the playbook** option, the **Target objects** is invoked. Select the objects from the **Observables** and **Assets** tabs for which you want to launch the playbook, and then click the **Apply and launch** button.

5. The **Playbook launched in test mode** popup appears. When the test run finishes, the **Playbook is completed in test mode** popup is shown. Click the **View results** link in this popup to display the panel that shows the result. If the test run succeeded, the **Playbook success** panel is invoked. If the test run failed, the **Playbook Error** panel is invoked.

   Test run results are saved in the **Response history** section of the playbook. Test run results are not shown in the general **Response history** section.

Launching playbooks and response actions

Launching playbooks

Depending on your needs, you can configure the way to launch the playbook. You can select one of the following operation modes during the playbook creation:

- **Auto**. Select this operation mode if you want to automate the launch of playbook and response actions.

  Playbooks in this mode help automate threat response, and also reduce the time it takes to analyze alerts and incidents.

- **Training**. Select this operation mode if you want to check if the playbook is configured correctly.

  Playbooks in this mode will not be launched automatically when a corresponding alert or incident is detected. Instead, the playbook requests the user's approval to launch.

- **Manual**. Select this operation mode if you want to launch the playbook manually only.

  Playbooks in this mode have no trigger, so you can launch such playbooks for any alert or incident, depending on the selected playbook scope. For more details, see Launching playbooks manually.

You can also change the operation mode of the existing playbook. For more details, see Editing playbooks.

Launching response actions

Response actions can be launched manually, automatically within a playbook, or can be configured to request the user's approval before launching within the playbook. By default, manual approval of the response action is disabled.

For more details on how to configure the manual approval of a response action launched within the playbook, see Configuring manual approval of response actions.

Launching playbooks manually

Kaspersky Next XDR Expert allows you to manually launch any playbook that matches alerts or incidents you want to respond to.

To launch a playbook manually, you must have one of the following roles: Main administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

You can also launch a playbook for observables and assets if you have specified these objects when creating the playbook and when launching it.

Launching a playbook for an alert

To launch a playbook manually for an alert:

1. In the main menu, go to **Monitoring** → **Alerts**.
2. In the table of alerts, click the link with the ID of the alert for which you want to launch the playbook.
3. In the **Alert details** window that opens, click the **Select playbook** button.

   The **Select playbook** window opens.
4. In the list of playbooks that match the alert, select the playbook you want to launch, and then click the **Launch** button.

   If the selected playbook is already running for this alert, in the **Monitoring** window that appears, do one of the following:

   - If you want to wait until the current playbook instance is completed, click the **Wait and launch** button.

     The new playbook instance will be launched after the current one is completed.
   - If you want to launch a new playbook instance immediately, click the **Terminate and launch a new one** button.

     The current playbook instance will be terminated and the new one will be launched.
   - If you want to cancel the new playbook launch, click the **Close** button (×).

   > If the selected playbook already has the status **Awaiting approval**, after manual launch, the playbook status will change to **Launched**.

The playbook is launched for the selected alert. After the playbook is completed, you will receive a notification.

Launching a playbook for an incident

To launch a playbook manually for an incident:

1. In the main menu, go to **Monitoring** → **Incidents**, and then select the **XDR incidents** tab.
2. In the table of incidents, click the link with the ID of the incident for which you want to launch the playbook.
3. In the **Incident details** window that opens, click the **Select playbook** button.

   The **Select playbook** window opens.
4. In the list of playbooks that match the incident, select the playbook you want to launch, and then click the **Launch** button.

If the selected playbook is already running for this incident, in the **Monitoring** window that appears, do one of the following:

- If you want to wait until the current playbook instance is completed, click the **Wait and launch** button.

  The new playbook instance will be launched after the current one is completed.

- If you want to launch a new playbook instance immediately, click the **Terminate and launch a new one** button.

  The current playbook instance will be terminated and the new one will be launched.

- If you want to cancel the new playbook launch, click the **Close** button (×).

> If the selected playbook already has the status **Awaiting approval**, after manual launch, the playbook status will change to **Launched**.

The playbook is launched for the selected incident. After the playbook is completed, you will receive a notification.

Launching playbooks for objects specified by users

You can specify observables and assets for which a playbook must run. You have to create a playbook with the following settings:

- In the **Scope** list, select **Alert** or **Incident**.
- In the **Operation mode** list, select **Manual**.
- In the visual editor work area, when Configuring playbook execution steps in visual editor, use jq expressions to specify the objects (observables or assets) for which you want the playbook to launch. These objects will be the input to the playbook when it is launched.

> If you do not specify the objects in the playbook algorithm and only select them before launching the playbook, these objects will be ignored.

After the playbook is created, you can launch it for the selected objects.

> To do this, you must have one of the following XDR roles: Main administrator, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

To launch a playbook for the selected objects:

1. In the main menu, go to the **Monitoring** section, and then in the **Alerts** or **Incidents** section, click the ID of the alert or incident from which you want to launch the playbook.
2. In the details window that opens, click the **Select playbook** button.

   The **Select playbook** window opens.
3. Select the **Select target objects before launching the playbook** option, and then click the **Launch** button.

4. In the **Target objects** window that opens, select the objects from the **Observables** and **Assets** tabs for which you want to launch the playbook, and then click the **Apply and launch** button.

   The playbook is launched for the objects you selected.

You can view the result of the playbook from the **History** tab in the alert or incident details, from the playbook details or playbooks table by clicking the **View history** button, and from the Response history section.

For example, you write a script that is called during the executeCustomScript response action. When creating a playbook, in the visual editor work area, you add the **Run custom script** execution step with the playbook input data. Then, you have to run the script for an observable with an IP type that you select when launching the playbook. The script uses the IP address that you selected as a parameter:

```
{

"dslSpecVersion": "1.1.0",

"version": "1",

"actionSpecVersion": "1",

"executionFlow": [

{

"action": {

"function": {

"type": "executeCustomScript",

"params": {

"commandLine": "./script.py",

"commandLineParameters": "${ \"-ip \" + ([.input.observables[] | select(.type == \"ip\")]
| map(.value) | join(\",\")) }",

"workingDirectory": "/folder/with/script"

}

},

"onError": "stop"

}

},

{

"action": {

"function": {
```

```
"type": "updateBases",

"params": {

"wait": false

},

"assets": "${ [.input.assets[] | select(.Type == \"host\") | .ID] }"

}

}

}

]

}
```

Several objects will be an input to the playbook, and the list of IP addresses separated with commas must be an input to the script:

```
{

"input": {

"observables": [

{

"type": "ip",

"value": "127.0.0.1"

},

{

"type": "ip",

"value": "127.0.0.2"

},

{

"type": "md5",

"value": "29f975b01f762f1a6d2fe1b33b8e3e6e"

}

],
```

```
"assets":[

{

"AttackerOrVictim": "unknown",

"ID": "c13a6983-0c40-4986-ab30-e85e49f98114",

"InternalID": "6d831b04-00c2-44f4-b9e3-f7a720643fb7",

"KSCServer": "E5DE6B73D962B18E849DC0BF5A2BA72D",

"Name": "VIM-W10-64-01",

"Type": "host"

}

]

}
```


After jq expressions perform calculations on the playbook operational data, the following information is passed as command line parameters:

```
-ip 127.0.0.1,127.0.0.2
```

For a playbook expecting input data, if you specified different types of objects when creating the playbook and when launching it, or if you did not select the **Select target objects before launching the playbook** option, the playbook will finish with one of the following results:

- An error will occur because the playbook did not receive input data.
- The action will not be performed because the playbook contains a condition or a loop that is based on the input data.
- The result will depend on the response of the application, or service, or script that performs the action.

Launching playbooks in the Training operation mode

The **Training** operation mode allows you to check if the playbook is configured correctly. This can be helpful if you are planning to change the playbook operation mode to **Auto**.

All playbooks in the **Training** operation mode request the user's approval to launch.

> To launch a playbook in the **Training** operation mode, you must have one of the following roles: Main administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

In the **Training** operation mode, the playbook is triggered by alerts or incidents, but requires manual approval before launch. You can test launching the playbook in the **Training** operation mode in one of the following ways:

- Create an alert or incident that matches the playbook trigger.
- Edit an alert or incident that matches the playbook trigger. The alert or incident must be in a status other than **Closed**.

> Editing an alert or incident triggers the playbook only if the trigger is set to respond to manual changes. By default, manual changes to an alert or incident are ignored. To make the playbook respond to manual changes, the trigger must use the `event.manual` function. For details, refer to Playbook trigger.

When one of the above actions is completed, the playbook requests the user's approval to launch. For more information on how to approve the playbook, see Approving playbooks or response actions.

> The manual approval for launching the playbook is limited to 60 minutes.

Configuring manual approval of response actions

Kaspersky Next XDR Expert allows you to configure manual approval of a response action launched within a custom playbook. By default, manual approval of the response action is disabled.

> Before configuring manual approval, make sure that email notifications for tenants are configured and the email address of the approver is specified.

> We recommend that you configure manual approval of the following response actions: moving devices to another administration group, moving files to quarantine, enabling and disabling network isolation, responding on accounts through Active Directory, and data enrichment.

To configure manual approval of a response action:

1. In the main menu, go to **Monitoring** → **Playbooks**.
2. Open the playbook for editing by doing one of the following:
   - Click the name of the playbook that you want to edit. In the **Playbook details** window that opens, click the **Edit** button.
   - Select the playbook from the list, and then click the **Edit** button.

     If you select more than one playbook, the **Edit** button will be disabled.

   The **Edit playbook** window opens.

3. In the **Algorithm** section, specify one of the following parameters for the response action for which you want to enable the manual approval:

   - To enable the manual approval of a response action with the default approval time, specify the following parameter:

     ```
     "manualApprove": true
     ```

     By default, the approval time is 60 minutes.

   - To enable the manual approval of a response action with an adjustable approval time, specify the following parameter:

     ```
     "manualApprove": {"timeout": "period"}
     ```

     where `"period"` is an adjustable approval time.

     You can configure the approval time in hours (h) and/or minutes (m), for example:

     ```
     "manualApprove": {"timeout": "20h"}
     "manualApprove": {"timeout": "2h30m"}
     ```

   - To enable the manual approval of a response action with notifications sent to the email address of the approver, specify the following parameter:

     ```
     "emailNotifications": {
      "enabled":true
     }
     ```

   - To enable the manual approval of a response action with a notification that is sent to the email address of the approver after a certain period, specify the following parameter:

     ```
     "manualApprove": {
       "emailNotifications": {
         "enabled": true,
         "delay": "period"
       }
     ```

     where `"period"` is an adjustable sending time.

     You can configure the sending time in minutes (m), for example:

     ```
     "delay": "20m"
     ```

4. Click the **Save** button.

   Manual approval of a response action is configured. Email notifications with a request to approve the response action will be sent to the email specified in the user account properties.

You can view requests for approval of response actions in the **Approval requests** section.

Approving playbooks or response actions

All playbooks in the **Training** operation mode require a user's approval. You can also configure manual approval of response actions launched within the playbook.

To approve or reject a playbook launch, you must have one of the following roles: Main administrator, Tier 1 analyst, Tier 2 analyst, Tenant administrator.

To approve or reject a response action launch, you must have one of the following roles: Main administrator, Approver, Tenant administrator.

If there are playbooks or response actions that are awaiting a user's approval, a notification appears at the top of the Open Single Management Platform Console. Additionally, if the user's approval is required for the response action launch, an email notification is sent to the email address within the time period specified in the algorithm of the playbook.

Viewing the list of playbooks and response actions

To view the list of playbooks and response actions that are awaiting approval, do one of the following:

- Click the **View approval requests** link at the top of the Open Single Management Platform Console.
- Follow the link in the notification that is sent to your email address.

The **Approval requests** pane with the table that contains the full list of approval requests opens.

The **Approval requests** table contains the following columns:

- **Time**. Date and time when a playbook or a response action requested a user's approval.
- **Approval due date**. Date and time by which the user must approve or reject the playbook or the response action. If the user has not approved the playbook or the response action by this time, the launch is canceled.
- **Playbook**. Name of a custom or predefined playbook that requests a user's approval.
- **Response action**. Actions that are launched within playbooks.
- **Assets**. Number of the assets for which the playbook or response action is launched. You can view the list of assets for which the user's approval is requested by clicking the link with the number of the assets.
- **Response parameters**. Response action parameters that are specified in the response action or playbook algorithm.
- **Alert or incident ID**. ID that contains a link to the alert or incident details.

Approving and rejecting playbooks

To approve or reject playbooks:

1. In the notification at the top of Open Single Management Platform Console, click the **View approval requests** link.

   A notification with the **View approval requests** link is displayed only if there is a playbook that is waiting for a user's approval.

2. In the **Approval requests** pane that opens, select one or more playbooks, and then do one of the following:

   - To approve the launching of a playbook, click the **Approve** button.

     After that, the playbook is launched. The action status in the **Response history** sections changes to **In progress**.

   - To decline the launching of a playbook, click the **Reject** button.

     After that, the launching of the playbook is canceled. The action status in the **Response history** sections changes to **Rejected**.

3. Click the **Close** button (×) to close the **Approval requests** pane.

After approving or rejecting playbooks, you can view their statuses in the **Response history** section.

Approving and rejecting response actions

To approve or reject response actions:

1. In the notification at the top of Open Single Management Platform Console, click the **View approval requests** link.

   A notification with the **View approval requests** link is displayed only if there is a response action that is waiting for a user's approval.

   The **Approval requests** pane opens.

2. In the **Approval requests** pane that opens, in the **Assets** column, click the link with the number of assets.

   The **Assets to approve** pane that contains the full list of assets opens.

3. Check the list of assets for which the manual approval is required, and then do one of the following:

   - To approve the launch of a response action for assets, select one or more assets you need, and then click the **Approve** button. After that, the response action for the selected assets is launched.

   - To decline the launch of a response action for assets, select one or more assets you need, and then click the **Reject** button. After that, the launch of the response action for the selected assets is canceled.

4. Click the **Close** button (×) to close the **Assets to approve** pane.

5. Click the **Close** button (×) to close the **Approval requests** pane.

After approving or rejecting response actions, you can view their statuses in the **Response history** section.

Enrichment from playbook

After you configure integration between Kaspersky Next XDR Expert and Kaspersky TIP, you can obtain information about the reputation of observables related to an alert or incident from Kaspersky TIP or Kaspersky OpenTIP, and then enrich the obtained data.

> You can obtain information only for observables with the following types: domain, URL, IP, MD5, SHA256.

You can configure data enrichment to run automatically. To do this, when creating or editing a playbook, in the visual editor work area you must add the Configuring playbook execution steps in visual editor, and specify the following:

1. Limit for data returned by Kaspersky TIP or Kaspersky OpenTIP, if necessary.

   You can specify one of the following values:

   - All records
   - Top100

     > This value is set by default.

2. Observable for which the playbook requests data from Kaspersky TIP or Kaspersky OpenTIP.

   > Data enrichment is performed by using the service specified in the integration with Kaspersky Threat Intelligence Portal.

In the playbook algorithm, you can use the output enrichment parameters that are displayed in the fields that Kaspersky TIP returns.

You can view the enrichment result for all observables related to an alert or incident in one of the following ways:

- From the alert or incident details
- From a response history
- From a playbook

To view an enrichment result:

1. In the main menu, go to the **Monitoring** section, and then do one of the following:
   - If you want to view the result from an alert or incident details, go to the **Alerts** or **Incidents** section, and then click the ID of the alert or incident for which the enrichment was performed. In the window that opens, go to the **History** tab, and then select the **Response history** tab.
   - If you want to view the result from a response history, go to the **Response history** section.
   - If you want to view the result from the playbooks table, go to the **Playbooks** section, and do one of the following:
     - Select the playbook for which the enrichment was performed. In the toolbar, click the **View history** button, and then select **View response history**.
     - Click the name of the playbook for which the enrichment was performed. In the window that opens, click the **View history** button, and then select **View response history**.
2. In the **Action status** column, click the status of the playbook for which you want to view the enrichment result.

You can also obtain the information from Kaspersky TIP, and then enrich data manually on the **Observables** tab in alert or incident details.

Viewing response history

The **Response history** section allows you to view the detailed response history for all detected alerts and incidents. Note that if an alert or incident is deleted, the response history for this alert or incident is not displayed.

> To view a response history, you must have one of the following roles: Main administrator, Tier 1 analyst, Tier 2 analyst, SOC manager, Junior analyst, Approver, Auditor, Tenant administrator.

To view a response history, in the main menu, go to **Monitoring** → **Response history**. The table that contains the response history for all alerts and incidents opens.

By default, the table is sorted according to the time the playbook or response action was launched. The response actions in the playbooks are sorted according to their order in the playbook algorithm.

The toolbar in the upper part of the table allows you group and filter the data in the table as follows:

- Click the settings icon (⚙), and then select the columns to be displayed in the table.
- Click the filter icon ( ), and then specify and apply the filter criterion in the invoked menu.

  > When you apply the filter criterion for the **Action status** column, the table displays the manually launched responses whose status contains the selected value and the playbooks that include the response actions whose status contains the selected value. It means that only the response actions of the playbook that meet the filter criterion will be displayed.

Table search is available for all columns except **Action status** and **Playbook status**. To filter data in these columns, click the column header and select the required status from the drop-down list.

The table contains the following columns:

- **Actions**. Response action or playbook name.
- **Response parameters**. Response action parameters that are specified in the response action or playbook algorithm.
- **Start**. Date and time the playbook or response action was launched.
- **End**. Date and time the playbook or response action was completed.
- **Alert or incident ID**. ID that contains a link to the alert or incident details.
- **Launched by**. Name of the user who launched the playbook or response action.
- **Action status**. Execution status of the response action. The following values can be shown in this column:
  - **Awaiting approval**—Response action awaiting approval for launch.
  - **In progress**—Response action is in progress.
  - **Success**—Response action is completed without errors or warnings.
  - **Warning**—Response action is completed with warnings.
  - **Error**—Response action is completed with errors.
  - **Terminated**—Response action is completed because the user interrupted the execution.
  - **Approval time expired**—Response action is completed because the approval time for the launch has expired.
  - **Rejected**—Response action is completed because the user rejected the launch.
- **Playbook status**. Execution status of the playbook. The following values can be shown in this column:
  - **Awaiting approval**—Playbook awaiting approval for launch.
  - **In progress**—Playbook is in progress.
  - **Success**—Playbook is completed without errors or warnings.
  - **Warning**—Playbook is completed with warnings.
  - **Error**—Playbook is completed with errors.
  - **Terminated**—Playbook is completed because the user interrupted the execution.
  - **Approval time expired**—Playbook is completed because the approval time for the launch has expired.
  - **Rejected**—Playbook is completed because the user rejected the launch.

You can click the **Playbook status** value or the **Action status** value to open the window with the result of the playbook or the response action launch. The Launch ID displayed on the **Debug data** tab can be used by Technical Support to download the data necessary for analysis in case of the execution errors. By default, the data is stored for 72 hours.
If the status is **In progress**, you can view the Launch ID by hovering the mouse cursor over the icon next to the status.

- **Assets**. Number of the assets for which the playbook or response action is launched. You can click the link with the number of the assets to view the asset details. The field is empty, if the playbook or response action does not involve assets.

- **Tenant**. Name of the tenant to which the playbook belongs.

- **Approver**. Name of the user who approved or rejected the playbook or response action launch.

  By default, this column is hidden. To display the column, click the settings icon (⚙), and then select the **Approver** column.

- **Approval time**. Date and time the playbook or response action launch was approved or rejected. This column is not displayed by default.

  By default, this column is hidden. To display the column, click the settings icon (⚙), and then select the **Approval time** column.

Predefined playbooks

Kaspersky Next XDR Expert provides ready-to-use predefined playbooks that are created by Kaspersky experts. Predefined playbooks are based on KUMA correlation rules. For more information on the KUMA correlation rules included in the distribution kit, see Correlation rules.

You can find predefined playbooks in the **Playbooks** section. Such playbooks are marked with the tag **Predefined** and the [KL] prefix in the name.

Note that you cannot edit the parameters of a predefined playbook, except for the **Operation mode** and the **Action on launched playbook instances** fields. If you want to edit other parameters of a predefined playbook, you need to duplicate the playbook, and then use it as a template to create a custom playbook. For details, refer to Customizing playbooks.

Before using the predefines playbooks, you must do the following in KUMA:

- Configure the enrichment rule settings for the event enrichment with the **Event** type selected as the **Source kind** setting. Specify the **VictimUserID** and **AttackerUserID** values in the **Target field**.

- Configure enrichment in KUMA to get Windows Event Log.

Predefined playbooks cannot be deleted.

Predefined playbooks belong to the parent tenant and are inherited by all child tenants.

[KL] P001 "Creation of executable files by office applications"

This playbook contains the Responding through KASAP response action, and can be used only as a template. If you want to launch the playbook, click the **Duplicate and edit** button. In the window that opens, in the **Assign KASAP group** action, specify the KASAP group ID.

Before using the playbook, you must configure enrichment in KUMA to get Windows Event Log.

By default, the playbook launches the response actions for all users in the alert. If you want the playbook to launch the response actions only for the victim account, you can do the following:

1. In KUMA, configure the enrichment rule settings. For the event enrichment that has the **Event** type selected as the **Source kind** setting, specify the **VictimUserID** value in the **Target field**.

2. In the **Assign KASAP group** action, specify `and .IsVictim` in the **Assets** field, as shown below:

   ```
   [ alert.Assets[] | select(.Type == \"user\" and .IsVictim) | .ID]
   ```

The **[KL] P001 "Creation of executable files by office applications"** predefined playbook allows you to prevent an attacker from using office applications, for example, to perform a phishing attack when a user opens an infected document, and then the document creates an executable file and executes it.

The alert that triggers the playbook is created according to the *Creation of executable files by office applications* correlation rule. This rule helps to detect the creation of files with suspicious extensions such as scripts and executable files on behalf of office applications.

The playbook trigger contains the following expression:

```
[.OriginalEvents[] | .ExternalID == "R350"] | any
```

During execution, this playbook launches the following response actions:

1. Responding through Active Directory, and then resetting the passwords of both the attacker and the victim accounts.

   If an error occurs during the execution of the response action, the playbook is terminated.

2. Responding through KASAP, and then assigning an information security course to the account.

   If an error occurs during the execution of the response action, the execution of the playbook will continue.

The playbook algorithm contains the following sequence of response actions:

```
{
    "dslSpecVersion": "1.1.0",
    "version": "1",
    "actionSpecVersion": "1",
    "executionFlow": [
```

```
        {
            "action": {
                "function": {
                    "type": "resetLDAPPassword",
                    "assets": "${[ alert.Assets[] | select(.Type == \"user\") | .ID]}"
                },
                "onError": "stop"
            }
        },
        {
            "action": {
                "function": {
                    "type": "assignKasapGroup",
                    "assets": "${[ alert.Assets[] | select(.Type == \"user\") | .ID]}",
                    "params": {
                        "groupId": "SET KASAP GROUP ID"
                    }
                },
                "onError": "continue"
            }
        }
    ]
}
```

[KL] P002 "Windows Event Log was cleared"

> By default, this playbook operates in the **Manual** operation mode. We do not recommend switching this playbook to the **Auto** or the **Training** operation mode.

Before using the playbook, you must do the following in KUMA:

- Configure the enrichment rule settings for the event enrichment that has the **Event** type selected as the **Source kind** setting. Specify the **AttackerUserID** value in the **Target field**.
- Configure enrichment in KUMA to get Windows Event Log.

The **[KL] P002 "Windows Event Log was cleared"** predefined playbook allows you to prevent an attacker from clearing the Windows Event Log, because the log contains sufficient telemetry for an investigation of the attacker's malicious activity.

The incident that triggers the playbook contains one or several alerts created according to the *Windows Event Log was cleared* correlation rule. This rule helps to detect when Windows logs are cleared or deleted by using the wevutil utility, the user interface, or PowerShell commands. To enable the creation of the incident, you have to configure segmentation rules.

The playbook trigger contains the following expression:

```
[.Alerts[] | .OriginalEvents[] | .ExternalID == "R050"] | any
```

During execution, this playbook launches the Responding through Active Directory response action, and then blocks the account of the attacker.

> If an error occurs during the execution of the response action, the playbook is terminated.

> If one or several alerts in the incident are generated by another correlation rule, the playbook does not apply to those alerts.

The playbook algorithm contains the following sequence of response actions:

```
{
    "dslSpecVersion": "1.1.0",
    "version": "1",
    "actionSpecVersion": "1",
    "executionFlow": [
        {
            "action": {
                "function": {
                    "type": "blockLDAPAccount",
                    "assets": "${[ incident.Alerts[] | select(.OriginalEvents[] |
.ExternalID == \"R050\") | .Assets[] | select(.Type == \"user\" and .IsAttacker) | .ID]}"
                },
                "onError": "stop"
            }
        }
    ]
}
```

[KL] P003 "Suspicious child process from wmiprvse.exe"

Before using the playbook, you must do the following in KUMA:

- Configure the enrichment rule settings for the event enrichment that has the **Event** type selected as the **Source kind** setting. Specify the **AttackerUserID** value in the **Target field**.
- Configure enrichment in KUMA to get Windows Event Log.

The **[KL] P003 "Suspicious child process from wmiprvse.exe"** predefined playbook allows you detect pairs of parent and child processes that deviate from the norm and must be viewed as suspicious.

The alert that triggers the playbook is created according to the *R297_Suspicious child process from wmiprvse.exe* correlation rule. This rule helps to detect the launch of suspicious processes on behalf of wmiprvse.exe.

The playbook trigger contains the following expression:

```
[.OriginalEvents[] | .ExternalID == "R297"] | any
```

During execution, this playbook launches the following response actions:

1. Responding through Active Directory, and then blocks the account of the attacker.
2. Terminating the process on the device that is registered in the alert.

3. Running a malware scan, and then a full scan is performed on the device where the alert is detected.

> By default, network drives are not scanned, to avoid overloading the system. If you want to scan the network drives, you have to duplicate this playbook, and then in the **Run malware scan** action, enable the **Scan network drives** toggle switch or set the `allowScanNetworkDrives` parameter to `true` in the code editor.

The playbook algorithm contains the following sequence of response actions:

```
{
    "dslSpecVersion": "1.1.0",
    "version": "1",
    "actionSpecVersion": "1",
    "executionFlow": [
        {
            "action": {
                "function": {
                    "type": "blockLDAPAccount",
                    "assets": "${[ alert.Assets[] | select(.Type == \"user\" and
.IsAttacker) | .ID]}"
                },
                "onError": "stop"
            }
        },
        {
            "loop": {
                "input": "${ [alert.OriginalEvents[] | [select(.DestinationProcessName !=
null and .DestinationProcessName != \"\")][] | .DestinationProcessName] }",
                "onError": "stop",
                "steps": [
                    {
                        "action": {
                            "function": {
                                "type": "killProcess",
                                "params": {
                                    "path": "${ .[0] }"
                                },
                                "assets": "${[ alert.Assets[] | select(.Type == \"host\")
| .ID]}"
                            }
                        }
                    }
                ]
            }
        },
        {
            "action": {
                "function": {
                    "type": "avScan",
                    "params": {
                        "scope": {
                            "area": "full",
                            "allowScanNetworkDrives": false
                        },
                        "wait": false
                    },
                    "assets": "${[ alert.Assets[] | select(.Type == \"host\") | .ID]}"
```

```
                },
                "onError": "stop"
            }
        }
    ]
}
```

If an error occurs during the execution of any response action, the playbook is terminated.

[KL] P004 "Sandbox Medium Detect"

The **[KL] P004 "Sandbox Medium Detect"** predefined playbook creates a rule to block execution of files assigned **Medium** alert severity by the Sandbox component during analysis.

The playbook trigger contains the following expression:

```
"event.new and .Severity == \"medium\" and any(.DetectionTechnologies[] == \"SB\"; .)"
```

ResponseFunction used by the playbook: `addFilePreventionRules`.

During execution, this playbook uses the Add Prevention Rule response action.

The playbook algorithm contains the following sequence of response actions:

```
{
  "version": "1",
  "dslSpecVersion": "1.1.0",
  "actionsSpecVersion": "1",
  "executionFlow": [
    {
      "action": {
        "function": {
          "type": "addFilePreventionRules",
          "assets": {
            "selector": "execution-tenant"
          },
          "params": {
            "files": "${[alert.ScannedFiles[] | select(any(.DetectionTechnologies[] == \"SB\"; .)) | .Hashes |
map(.Value)] | flatten}",
            "notify": true
          }
        },
        "onError": "stop"
      }
    }
  ]
}
```

[KL] P005 "Sandbox High Detect"

The **[KL] P005 "Sandbox High Detect"** predefined playbook creates a rule to block execution of files assigned **High** alert severity by the Sandbox component during analysis.

The playbook trigger contains the following expression:

```
"event.new and .Severity == \"high\" and any(.DetectionTechnologies[] == \"SB\"; .)"
```

ResponseFunction used by the playbook: `addFilePreventionRules`.

During execution, this playbook uses the Add Prevention Rule response action.

The playbook algorithm contains the following sequence of response actions:

```
{
  "version": "1",
  "dslSpecVersion": "1.1.0",
  "actionsSpecVersion": "1",
  "executionFlow": [
    {
      "action": {
        "function": {
          "type": "addFilePreventionRules",
          "assets": {
            "selector": "execution-tenant"
          },
          "params": {
            "files": "${[alert.ScannedFiles[] | select(any(.DetectionTechnologies[] == \"SB\"; .)) | .Hashes |
map(.Value)] | flatten}",
            "notify": true
          }
        },
        "onError": "stop"
      }
    }
  ]
}
```

Playbook trigger

The playbook trigger is a filter that allows you to select alerts or incidents for which a playbook must be launched. The filter (trigger) is applied to each object (alert or incident) individually and takes a single value: either `true` or `false`. A trigger consists of expressions in the jq language that processes structured data in the JSON format. For more information about jq expressions, refer to jq Manual.

In Kaspersky Next XDR Expert, gojq is used. It is an implementation of jq written in the go language, which has the following differences from jq:

- Mathematical functions are implemented in a more convenient way.
- Errors messages have a clear indication of where to fix your query.
- Integer calculations are more accurate.
- Functions that work incorrectly in jq are improved in gojq.

For more information about the differences between gojq and jq, refer to GitHub.

How to write a trigger

You can write a trigger in the **Trigger** block in visual editor when creating or editing a playbook.

Depending on the option you select in the **Scope** list when creating or editing a playbook, alert data model or incident data model is used.

> The names of parameters in a playbook trigger must be the same as in the data model. Note that elements of jq expressions are case-sensitive.

> To avoid overloading the system, it is not recommended to specify `OriginalEvents, Observables, Extra,` and `Alerts` data in the trigger.

When you start writing a trigger, the following suggestions can be displayed:

- Names of functions.
- Special values.
- Fields that are specified as object identifiers in accordance with the data model.

The suitable values are filtered and displayed in the list of suggestions when you start typing. For convenience, some suggestions contain a search string. For example, if you want to specify an incident type ID or incident status ID, you can search the corresponding record by the name, and the ID will be specified in the trigger.

> Note that default statuses (*New* and *Closed*) have the same ID in different workflows, which means that the trigger will run for all incidents with the specified status ID. To limit the scope of incidents for which the playbook will run, in the trigger you have to specify the incident status ID and the incident type.

The jq language also provides syntax highlighting and validation of jq expressions. If the trigger has invalid expressions, you cannot save the playbook.

When writing a trigger, basic syntax rules are used.

To refer to structure properties, you must use dot "`.`" and specify the attribute, for example:

- `.MITRETactics[]`—To view the array of MITRE tactics associated with all triggered IOA rules in the alert.
- `.MITRETactics[0]`—To view the first element from the MITRE tactics array.

To refer to child properties, you can either use the pipe (|) or the same combination without the pipe, for example:

- `.Assignee.Name`—The expression outputs the name of the user to whom the alert is assigned.
- `.MITRETactics[0].ID` or `.MITRETactics[0] | .ID`—The expression outputs the ID of the first MITRE tactic.

To get a value, you have to use the following operators: ==, >, <, >=, <=, !=, for example:

- `.Assignee.Name == "user"`—The expression returns `true` if the alert is assigned to the user.
- `(.Severity == "high") and (.DetectSource == "KES")`—The expression returns `true` if the alert severity level is high and the source of data is Kaspersky Endpoint Security.

- `[ .DetectionTechnologies[] | . == "IOC" ] | any` —The expression returns `true` if the IOC detection technology is triggered.

- `.DetectionTechnologies | length > 1`—The expression returns `true` if more than one detection technology is triggered.

To enumerate values in an array of objects, you can use the `any` method, for example:

- `[.Assets[] | .Name == "W21H2-X64-3160"] | any`—The expression filters the alert where any element of the `Assets` array has the `W21H2-X64-3160` value in the `Name` field.

- `[.Observables[] | .Value == "127.0.0.1"] | any`—The expression filters the alert where any element of the `Observables` array has the `127.0.0.1` value in the `Value` field.

- `[.Assets[].ID]`—To output the array of IDs.

- `[.Assets[] | select(.AttackerOrVictim=="attacker") | .ID]`—To display an array of IDs for the assets filtered by the `AttackerOrVictim` field.

If you want to reuse calculations, specify a variable with `$`. For example, the expression `event.manual != true as $not_manual | [ .DetectionTechnologies[] | . == "IOC" ] | any and $not_manual` defines and uses the `$not_manual` variable that contains a flag that shows if the change is manual or not.

To work with dates, you can use the following functions:

- `now`—To get the current Unix time in seconds, for example, `now == 1690541520.537496.`

- `todate`—To convert the current Unix time to a date string, for example, `now | todate == "2023-07-28T10:47:36Z".`

- `fromdate`—To convert the date to seconds, for example:

  - `.CreatedAt | split(".")[0] + "Z"`—This command removes milliseconds and converts the string to the 2023-07-15T07:49:51Z format.

  - `(.CreatedAt | split(".")[0] + "Z") | fromdate == 1689407391`—The conversion to seconds is finished.

Jq uses iterators—an interface that provides access to elements of a collection, for example, an array, and allows you to navigate through them. Iterators are always the result of calculation. The difference is in the number of elements that the iterator contains. In Kaspersky Next XDR Expert, an iterator must have only one element; the other cases are considered an error.

To write a correct trigger, you have to wrap an iterator into square brackets (`[ ... ]`). For example, the `.DetectionTechnologies[] == "IOC"` trigger will cause an error because it returns an iterator with two elements. The correct trigger must have the following form: `[ .DetectionTechnologies == "IOC" ] | any`, where first you have to use `[]` to wrap the result of the comparison into an array, and then process it with the `any` method that returns `true` if at least one element of the array is `true`. Otherwise, `false` is returned.

When the trigger runs

The search for a suitable playbook starts when one of the following triggering events occurs:

- New alert/incident is created.
- Any field of an active alert/incident is changed.
- When creating or editing a playbook, the user selected the **Launch the playbook for all matching alerts or incidents. Note that the system may be overloaded** check box.

The following types of alert change events are supported:

- Assigning or removing an analyst.
- Changing an alert status.
- Changing basic events.
- Linking or unlinking an alert to or from an incident.
- Changing the value in the `ExternalReference` field.

The following types of incident change events are supported:

- Assigning or removing an analyst.
- Changing an incident status.
- Changing basic events.
- Linking or unlinking an alert to or from an incident.
- Changing an incident name.
- Changing an incident description.
- Changing an incident priority.
- Changing the value in the `ExternalReference` field.
- Merging incidents.

The alert/incident structure does not contain any data about the alert/incident changes. This data is transferred in additional information. If in a playbook trigger you want to refer to the changes, use the event function without arguments.

By default, manual changes to an alert or incident details are ignored. If you want a playbook to launch for manual changes, you have to use the `event.manual` function in the trigger, for example:

- `event.manual and ([ event.updateOperations[] | . == "alertReopened" ] | any)`—The trigger works only if the alert is manually reopened.
- `[ event.updateOperations[] | . == "alertLinkedWithIncidentBySystem" ] | any`—The trigger works only if the alert is automatically linked to an incident.

- `event.manual != null and (([ event.updateOperations[] | . == "alertChangedToNew" ] | any) | not)`—The trigger works if the alert status is changed to any status other than *New*, either manually or automatically.

- `event == null and .Status == "inIncident"`—The trigger works for all alerts with the *In incident* status, but only when the playbook is changed, not the alert.

If necessary, you can test examples of jq expressions, apply filters, and then view the results in the Jq playground service.

Playbook algorithm

Kaspersky Next XDR Expert allows you to respond to alerts and incidents manually or automatically by using playbooks. Responding to alerts or incidents may consist not of a single action, but of a whole set of steps and parameters. These steps depend on the specified conditions, the alert or incident data, and the results of previous response actions.

The playbook algorithm allows you to specify the sequence of response actions, the necessary conditions, and the required impact on the target objects, in the JSON format. The playbook algorithm steps are performed sequentially. You can specify the playbook algorithm when creating or editing a playbook.

After launch, the playbook obtains all the alert or incident data, and places them in global data. The playbook uses the following data:

- Global data

  Global data is readable at any step of the playbook. Global data contains information about the alert or incident for which the playbook was launched.

  > You cannot edit global data by using a playbook, or by changing alert or incident data. Global data remains unchanged for the entire lifetime of the playbook instance.

- Operational data

  Operational data is transferred between the steps of the playbook. You can manage operational data by using jq expressions, which are specified in the `input` and `output` parameters.

- Local data

  Local data is limited to a specific step. You can manage local data by using the `input` (local data generation) and `output` (generation of operational data from local data) parameters.

How to write an algorithm

The playbook algorithm is written in the JSON format and consists of two main parts:

- General information on the playbook:

  - Name (`name`)

  - Description (`description`)

  - Scope (`inputType`)

    - Transformation of the input data of the playbook (`input`)
    - Transformation of the output data of the playbook (`output`)
    - Playbook execution timeout (`playbookRunTimeout`)
    - Timeout policies that can be applied at specific steps (`timeouts`)
    - Playbook version (`version`)
    - DSL schema version (`dslSpecVersion`)
    - Response action schema version (`actionsSpecVersion`)
- Playbook execution steps (`executionFlow`).

The following parameters are required when writing the algorithm:

- `dslSpecVersion`. The required value: 1.1.0.
- `actionsSpecVersion`
- `version`
- `executionFlow` (at least one execution step)

> Each execution step has its own required fields.

If you try to save a playbook without filling in the required fields, an error will appear.

The playbook algorithm is case-sensitive. To use the asset data of the alert, you need to capitalize the `Assets` parameter. For example: `alert.Assets[]`. However, to use asset data in the input data when manually launching the playbook for target objects, do not capitalize the `assets` parameter. For example: `.input.assets[]`.

Depending on the scope you selected when creating or editing a playbook, you can use the alert data model or incident data model in expressions written in the jq language. To do that, write expressions with an `alert` or `incident` value (do not use dot "`.`" at the beginning of the value). For example:

```
"${[ alert.Assets[] | select(.Type == \"user\" and .IsAttacker) | .ID]}"
```

You can use alert or incident data in a jq expression at any execution step. The alert or incident data is only available in read mode. This data does not change during the operation of the playbook. If alert or incident data has changed after launching the playbook, it will not affect the playbook execution.

You also can use the jq expressions when using the playbook data in the algorithm. For more information about jq expressions, refer to jq Manual.

> If you use quotation marks in the jq expression, you need to escape these marks with backslashes. For example: `"${[ alert.Assets[] | select(.Type == \"user\" and .IsAttacker) | .ID]}"`.

> Backslashes that are not used to escape quotation marks must also be escaped by other backslashes. For example: `${\"add_firewall_rule --ip_address=\" + ([.input.observables[] | select(.type == \"ip\") | select(.value | test(\"^(10\\\\.|172\\\\.(1[6-9]|2[0-9]|3[01])\\\\.|192\\\\.168\\\\.|127\\\\.).*\") | not) | .value] | join(\",\"))}`.

If you want to launch the playbook for the specific object (observables or assets), use the `.input` parameter in the algorithm. These objects will be the input to the playbook when it is launched. For example:

```
"assets": "${ [.input.assets[] | select(.Type == \"host\") | .ID] }"
```

For details, refer to Launching playbooks for objects specified by users.

How to call hints

If you need a hint on the available fields when writing the algorithm, use quotation marks (`""`). A list of available fields appears.

To display hints on the alert or incident data,

in the jq expression, write `alert` or `incident` and include a dot "`.`" at the end.

> The correct hint appears if there are no errors in the above expressions. Otherwise, the list of available fields may be incorrect.

How to call suggestions

You can call suggestions when writing the playbook algorithm. The suggestions contain a search string and help you specify the field value quickly. To view the suggestions, use quotation marks (`""`). A list of suggestions appears.

The suggestions also allow you to search by name. However, when you select the required value, the name will be automatically changed to an ID in the algorithm. For details, refer to Editing incidents by using playbooks and Editing alerts by using playbooks.

If you select and then delete the parameter name specified between the quotation marks, suggestions for the parameter will not appear, even if you have specified a new parameter between the quotation marks.

To return to the suggestion mode for the parameter, do one of the following:

- Delete the quotation marks, and then add an opening double quotation mark. A closing double quotation mark will be added automatically, and a suggestion will appear.
- Type any character between the quotation marks, and then press **`Backspace`**. This will return you to the suggestion mode.

If you delete the parameter name character by character, you will not exit the suggestion mode, even if you delete the parameter name completely.

Example of the playbook algorithm

```
{
  "actionsSpecVersion": "1",
  "dslSpecVersion": "1.1.0",
  "version": "1",
  "playbookRunTimeout": "24h",
  "executionFlow": [
    {
      "action": {
        "function": {
          "type": "blockLDAPAccount"
          "assets": "${[ alert.Assets[] | select(.Type == \"user\" and .IsAttacker) | .ID]}"
        },
        "onError": "stop"
      }
    },
    {
      "loop": {
        "batchSize": 1,
        "input": "${ [alert.OriginalEvents[] | [select(.DestinationProcessName != null and
.DestinationProcessName != \"\")][] | .DestinationProcessName] }",
        "mode": "parallel",
        "onError": "stop",
        "steps": [
          {
            "action": {
              "function": {
                "type": "killProcess",
                "assets": "${[ alert.Assets[] | select(.Type == \"host\") | .ID]}",
                "params": {
                  "path": "${ .[0] }"
                }
              }
            }
          }
        ]
      }
    },
    {
      "action": {
        "function": {
          "type": "avScan",
          "assets": "${[ alert.Assets[] | select(.Type == \"host\") | .ID]}",
          "params": {
            "scope": {
              "allowScanNetworkDrives": false,
              "area": "full"
            },
            "wait": false
          }
        },
        "onError": "stop"
      }
    }
  ]
}
```

Playbook parameters

Parameter ID	Description
`name`	Playbook name. Specified by the system when creating or updating a playbook. If the value is set in the algorithm, it will be replaced by the system.
`description`	Playbook description. Specified by the system when creating or updating a playbook. If the value is set in the algorithm, it will be replaced by the system.
`version`	Playbook version. The minimum length is 1. This parameter is required.
`dslSpecVersion`	DSL schema version. The minimum length is 1. This parameter is required.
`actionSpecVersion`	Response actions schema version. The minimum length is 1. This parameter is required.
`playbookRunTimeout`	The maximum execution time of the playbook, including waiting in the queue. The maximum value is 48 hours (`48h`). You can configure the maximum execution time in hours (h) and/or minutes (m). By default, the value is `24h`.
`inputType`	Inbound object type. The possible values: `alert` or `incident`. The inbound object type is specified by the system when creating or updating a playbook. If the value is set in the algorithm, it will be replaced by the system.
`input`	A jq expression that could be used to transform or filter incoming data before executing a playbook.
`output`	A jq expression that could be used to modify the output of the playbook before execution.
`timeouts`	Timeout definitions.
`executionFlow`	Steps of the playbook execution. This parameter is required.

Execution step parameters

The array of execution step elements describes a playbook's logic. The execution steps are performed in the order described in the playbook. There are several types of execution steps:

- Action
- Loop
- Parallel
- Decision
- UpdateData

Action parameters

The `Action` parameters call the response function.

Parameter ID	Description
`function`	An object that defines a response action. For more information, refer to ResponseFunction parameters.

`filterProduct`	This parameter allows you to filter components to perform a response action. When requested, the component plug-ins are filtered by allowed and restricted components. For example, the parameter can be specified as follows: `"filterProduct": { "allowed": ["PRODUCT_NAME"] }`
`output`	This parameter allows you to edit the value returned by the response action, by using a jq expression and placing it in the playbook data (local or operational).
`timeout`	This parameter allows you to set timeouts for calling the response function. You can specify the name of the timeout policy set in the playbook or set timeout values manually. If the value is not specified, the default timeout is applied.

`manualApprove`	This parameter allows you to configure a manual approval of a response action. Possible values:
	• Boolean value:
	  • `true`—Manual approval is enabled with default parameters.
	  • `false`—Manual approval is disabled.
	• Object `ManualApprove`.

`onError`	This parameter defines the behavior when an error occurs during the execution of a response action. Possible values: • `stop`—Defines the termination of the playbook in case of an error during the execution of the response action. • `continue`—Defines that the playbook execution will continue, even if one of the response actions completes with an error. In this case, the playbook launches the next response action specified in the algorithm. By default, the value is `stop`. Note that, if a system error occurs, the playbook execution completes with an error regardless of the specified value of the `onError` parameter.

Timeout policy

The timeout policy of execution steps. The system automatically determines the default timeout policy.

The default timeout policy can be reconfigured by using the default policy name. In this case, the new policy will be automatically applied to all execution steps.

Parameter ID	Description
`name`	Timeout policy name.
`scheduleToCloseTimeout`	The maximum execution time, including waiting in the queue and retries. The parameter is specified in the Go string format. If the value is not specified or 0, the value from the `playbookRunTimeout` field is used.

Output

The `output` parameter generates operational data at the end of a step, which will then be transferred to the next step. Specify the `output` parameter if you want to use the results of the current step of the playbook in the next step.

To avoid overloading the system, it is recommended to limit the data placed in the playbook data (local or operational).

Parameter ID	Description
`action`	This parameter defines whether the playbook data (local or operational) will be overwritten or merged. Possible values: • `merge` —The new data is merged with the current data. • `overwrite` —The current data is overwritten with new data.
`filter`	This parameter defines the jq expression for processing output data.

Manual approve

Parameter ID	Description
`timeout`	The timeout for manual approval in minutes. The minimum value is 10 minutes (`10m`), the maximum is 180 minutes (`180m`). By default, the value is 60 minutes (`60m`).
`emailNotifications`	This parameter allows you to configure the sending of email notifications.

Email notification settings

Parameter ID	Description
`enabled`	Flag for enabling email notifications.
`delay`	This parameter defines the delay before sending the email notification. The value is specified in minutes. The minimum value is 5 minutes (`5m`), the maximum is 30 minutes (`30m`). By default, the value is 10 minutes (`10m`).

Loop

Before specifying the `Loop` parameter, make sure that the `aggregate` parameter is also specified in the playbook algorithm.

The `Loop` parameters are used to split the array of incoming data by elements and to perform various actions on the elements.

Parameter ID	Description
`input`	A jq expression for composing an array or referencing an array.
`aggregate`	This parameter allows you to configure aggregation rules by using a jq expression.
`output`	Configuring how to apply the output data to the current playbook data. Possible values: • String constant: `merge` or `overwrite`. • Object `Output`.
`mode`	Loop operation mode. Possible values: • `parallel` —Defines that all elements are processed in parallel. The number of threads is controlled by the interpreter. • `sequence` —Defines that all elements are processed sequentially. By default, the value is `parallel`.
`batchSize`	This parameter allows you to specify the number of array elements that will be processed in one loop or one parallel thread. You can use this parameter if the plug-in function limits the number of input elements. For example, if a plug-in function can handle no more than 10 elements in one loop, you can specify the following parameter value: `batchSize=10`. By default, the value is `1`.

`onError`	This parameter defines the behavior when an error occurs in one of the branches. Possible values:
	• `stop` —Defines the termination of all branches if an error has occurred. The other branches will continue to run.
	  • If `mode=sequence`, after an error occurs in one branch, all subsequent branches will be stopped.
	  • If `mode=parallel`, after an error occurs in one branch, all branches will continue to run independently of each other.
	• `continue` —Defines the stop of one of the branches where the error occurred. The other branches will continue to run.
	By default, the value is `stop`.

`steps`	Array of execution steps.

Parallel

> Before specifying the `Parallel` parameter, make sure that the `aggregate` parameter is also specified in the playbook algorithm.

The `Parallel` parameters are used to perform several actions on the data at the same time. Unlike Loop, `Parallel` transmits the same input data to different execution branches.

Parameter ID	Description
`input`	A jq expression for composing an array.
`aggregate`	This parameter allows you to configure aggregation rules by using a jq expression.
`output`	Configuring how to apply the output data to the current playbook data. Possible values: • String constant: `merge` or `overwrite`. • Object `Output`.
`onError`	This parameter defines the behavior when an error occurs in one of the branches. Possible values: • `stop` —Defines the termination of all branches if an error has occurred. The other branches will continue to run. • `continue` —Defines the stop of one of the branches where the error occurred. The other branches will continue to run. By default, the value is `stop`.
`branches`	Execution branches.

Branch

Parameter ID	Description
`name`	The name of the branch that is unique within `Parallel`.
`steps`	Array of execution steps.

Decision

The `Decision` execution step allows you to perform a step or set of steps according to a condition. Note that only the first verified condition is executed.

Parameter ID	Description
`conditions`	Array of conditions.

Condition

Parameter ID	Description
`condition`	A jq expression that contains execution conditions.
`steps`	Execution steps for the current branch.

UpdateData

The *UpdateData* parameter can be described either as a jq script with state change logic, or as an `Output` object.

ResponseFunction parameters

Parameter ID	Description
`action`	Response action name.
`params`	The parameter allows you to describe the parameters of a response action you want to launch. You can specify the parameter as a jq expression or as an object. Parameters of the response actions are described in the table below.
`assets`	The parameter allows you to use a jq expression or string array to specify a list of assets for which you want to launch a response action. The `assets` parameter is required for response actions with assets and is not applicable for response actions without assets.

Response action parameters

Response action name	Parameters

`updateBases`	Update databases response action. Possible parameters:
	• `wait`. Possible values:
	  • `true`
	  • `false`
	To launch this response action, you need to specify the `asset` parameter of the response function.

`avScan`	Run malware scan response action. Possible parameters:
	• `wait`. Possible values:
	  • `true`
	  • `false`
	• `scope`. Possible values:
	  • `full`—Perform a full scan of the device where the alert is detected.
	  • `critical`—Perform a scan of the kernel memory, running processes, and disk boot sectors.
	  • `selective`—Perform a scan of the specified files. To specify a path to the files, use the `path` parameter.
	• `allowScanNetworkDrives`. Possible values:
	  • `true`
	  • `false`
	  By default, the value is `false`.
	  This parameter is available only if you want to perform a full scan.
	  Note that scanning network drives can overload the system.
	• `path`—A jq expression or a string with a path to the files you want to scan. You can also specify multiple file paths.
	To launch this response action, you need to specify the `asset` parameter of the response function.

`moveHostsToAdministrationGroup`	Move to group response action. Possible parameters: • `group`—Open Single Management Platform administration group path. For examples, `HQ/OrgUnit1`. To launch this response action, you need to specify the `asset` parameter of the response function.
`quarantineFile`	Move to quarantine response action. Possible parameters: • `path`—Path to the file you want to quarantine. • `md5`—MD5 hash of the file. • `sha256`—SHA256 hash of the file. You can specify the response action parameters in one of the following ways: • Specify the full path to the file you want to quarantine. In this case, you do not need to specify an MD5 hash or a SHA256 hash. • Specify the file path and the file hash (MD5 or SHA256). To launch this response action, you need to specify the `asset` parameter of the response function.
`killProcess`	Terminate process response action. Possible parameters: • `pid`—Process identifier. • `path`—Path to the file you want to quarantine. • `md5`—MD5 hash of the file. • `sha256`—SHA256 hash of the file. To launch this response action, you need to specify the `asset` parameter of the response function.

`changeAuthorizationStatus`	Change authorization status response action. Possible parameter:
	• `authorized`. Possible values:
	  • `true`
	  • `false`
	To launch this response action, you need to specify the `asset` parameter of the response function.

`netIsolateOn`	Enable network isolation response action. Possible parameters:
	• `isolationTimeoutSec` —Network isolation period. You can specify this parameter in hours or days. The minimum value in hours is 1 hour, the maximum is 9999 hours.
	The network isolation period is specified in seconds.
	• `exclusions` —Exclusion rules. You can specify one or more exclusion rules.
	  • `remoteIPV4Address` —Network traffic from the specified IPv4 address will be excluded from the block. For example, `192.168.2.15`.
	  • `remoteIPV6Address` —Network traffic from the specified IPv6 address will be excluded from the block. For example, `2001:0db8:0000:0000:0000:ff00:0042`.
	  • `remotePortRange` —Interval of remote ports.
	  • `localPortRange` —Interval of local ports.
	If the `remotePortRange` and `localPortRange` are not specified, the exclusion rule applies to all ports.
	• `exclusionsConflictBehavior` —Defines the behavior if there is a conflict between different exclusion rules. Possible parameters:
	  • `replace`
	  • `skip`
	  • `fail`

`netIsolateOff`	Disable network isolation response action. To launch this response action, you need to specify the `asset` parameter of the response function.
`executeCommand`	Run executable file response action. Possible parameters: • `commandLine` —Command to run or a full path to the executable file. • `workingDirectory` —Path to the working directory. • `commandLineParameters` —Command-line parameters that you want to apply to the command. To launch this response action, you need to specify the `asset` parameter of the response function.
`addFilePreventionRules`	Add prevention rule response action. Possible parameters: • `md5` —MD5 hash array. • `sha256` —SHA256 hash array. To launch this response action, you need to specify the `asset` parameter of the response function.
`deleteFilePreventionRules`	Delete prevention rule response action. Possible parameters: • `md5` —MD5 hash array. • `sha256` —SHA256 hash array. To launch this response action, you need to specify the `asset` parameter of the response function.
`resetFilePreventionRules`	Delete all prevention rules. To launch this response action, you need to specify the `asset` parameter of the response function.
`assignKasapGroup`	Assign KASAP group response action. Possible parameters: `groupId` —KASAP group identifier. To launch this response action, you need to specify the `asset` parameter of the response function.
`addToLDAPGroup`	Add user to security group response action. Possible parameters: `groupDN` —Distinguished name (DN) of the LDAP group. To launch this response action, you need to specify the `asset` parameter of the response function.
`removeFromLDAPGroup`	Delete user from security group response action. Possible parameters: `groupDN` —Distinguished name (DN) of the LDAP group. To launch this response action, you need to specify the `asset` parameter of the response function.
`blockLDAPAccount`	Lock account response action. To launch this response action, you need to specify the `asset` parameter of the response function.
`resetLDAPPassword`	Reset password response action. To launch this response action, you need to specify the `asset` parameter of the response function.

`executeCustomScript`	Execution of custom scripts. Possible parameters: • `commandLine`—Command to run. • `commandLineParameters`—Command-line parameters that you want to apply to the command. • `stdIn`—Standard input stream. Use this parameter if a script requires some additional data from the standard input. • `workingDirectory`—Path to the working directory.

`iocsEnrichment`	Data enrichment. Possible parameters:
	• `observables`—A jq expression with an array of observables that you want to enrich.
	• `fullEnrichment`—Defines the number of records to be requested. Possible values:
	  • `true`—Request all records from the source.
	  • `false`—Request the top 100 records from the source.

Editing incidents by using playbooks

Kaspersky Next XDR Expert allows you to edit incidents manually or by using playbooks. When creating a playbook, you can configure the playbook algorithm to edit the incident properties.

To edit an incident by using a playbook, you must have one of the following roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

You cannot edit incidents that have the **Closed** status.

You can edit the following incident properties by using the playbook:

- Assignee
- Incident workflow status
- Incident type
- Comment
- Description
- Priority
- ExternalReference attribute
- Additional data attribute

Below are example expressions you can use in the playbook to edit incident properties. You can specify these expressions in the **Show code** section of the visual editor.

- **Assigning an incident to a user**

```
{
        "action": {
          "function": {
             "type": "assignIncident",
             "params": {
```

```
                "assignee": {
                   "id": "user_ID"
                              }
                        }
                    }
                }
            }
```

When you edit an assignee in the playbook algorithm, suggestions are displayed. For convenience, the suggestions contain a search string where you can search by name. If you want to specify an incident assignee, you can search the corresponding record by the user's name, and the ID will be specified in the algorithm.

- **Assigning an incident to a user group**

```
{
    "action": {
      "function": {
         "type": "assignIncident",
          "params": {
            "group": {
               "id": "GROUP_ID"
                              }
                        }
                    }
                }
            }
```

When you edit an assignee in the playbook algorithm, suggestions are displayed. For convenience, the suggestions contain a search string where you can search by name. If you want to specify an incident assignee, you can search the corresponding record by the name of a user group, and the ID will be specified in the algorithm.

- **Unassigning an incident from a user**

```
{
        "action": {
          "function": {
             "type": "assignIncident",
             "params": {
                 "assignee": {
                    "id": "nobody"
                              }
                        }
                    }
                }
            }
```

- **Unassigning an incident from a user group**

```
{
    "action": {
      "function": {
         "type": "assignIncident",
          "params": {
            "group": {
```

```
                "id": "nobody"
                                }
                            }
                        }
                    }
                }
```

- **Changing a status of the incident workflow**

To change the incident workflow status to **Open**:

```
{
    "action": {
      "function": {
         "type": "setIncidentStatus",
         "params": {
           "typeId": "af9dd279-fc30-4596-963b-942f79920375",
           "statusId": "4db36105-5223-4078-b72c-e9e9983b0987"
                            }
                        }
                    }
                }
```

To change the incident workflow status to **Closed**:

```
{
        "action": {
          "function": {
             "type": "setIncidentStatus",
             "params": {
                    "statusId": "INCIDENT_STATUS_ID",
                 "statusResolution": "truePositive"
                            }
                        }
                    }
                }
```

You can also specify the following values for the `statusResolution` parameter: `falsePositive` and `lowPriority`.

To change the incident workflow status to a custom status:

```
{
        "action": {
          "function": {
             "type": "setIncidentStatus",
             "params": {
                 "typeId": "22222222-2222-2222-2222-222222222222",
                 "statusId": "11111111-1111-1111-1111-111111111111"
                            }
                        }
                    }
                }
```

When you edit an incident workflow status in the playbook algorithm, suggestions are displayed. For convenience, the suggestions contain a search string where you can search by name. If you want to specify an incident workflow status, you can search the corresponding record by the name, and the ID will be specified in the algorithm.

- Changing the incident type

```
{
        "action": {
          "function": {
             "type": "setIncidentType",
             "params": {
                    "id": "INCIDENT_TYPE_UUID"
                          }
                     }
                }
          }
```

When you edit an incident type in the playbook algorithm, suggestions are displayed. For convenience, the suggestions contain a search string where you can search by name. If you want to specify an incident type, you can search the corresponding record by the name, and the ID will be specified in the algorithm.

- Adding a comment to an incident

```
{
        "action": {
          "function": {
             "type": "addCommentToIncident",
             "params": {
                 "text": "${ \"New comment for incident with ID: \\(incident.ID)\" }"
                          }
                     }
                }
          }
```

- Editing the incident description

```
{
   "action": {
     "function": {
        "type": "setIncidentDescription",
        "params": {
              "description": "${ incident.ID | tostring | \"New comment for incident with ID: \" + . }",
              "mode": "replace"
                        }
                  }
              }
         }
```

To append to the existing description, specify the `append` value for the `mode` parameter.

- Changing the incident priority

```
{
        "action": {
          "function": {
             "type": "setIncidentPriority",
             "params": {
```

```
                    "priority": "critical"
                          }
                     }
                }
           }
```

You can also specify the following values for the `priority` parameter: `high`, `medium`, `low`.

- **Editing the ExternalReference attribute**

```
{
    "action": {
      "function": {
         "type": "setIncidentExternalRef",
         "params": {
               "externalRef": "${ \"new extReference value\" }",
               "mode": "replace"
                         }
                   }
               }
          }
```

To append to the ExternalReference attribute, specify the `append` value for the `mode` parameter.

- **Editing the Additional data attribute**

```
{
        "action": {
          "function": {
             "type": "addIncidentAdditionalData",
             "params": {
                 "data": "${ {\"customKey\": \"customValue\"} }",
               "mode": "replace"
                         }
                   }
                }
          }
```

To append to the Additional data attribute, specify the `append` value for the `mode` parameter.

Editing alerts by using playbooks

Kaspersky Next XDR Expert allows you to edit alerts manually or by using playbooks. When creating a playbook, you can configure the playbook algorithm to edit the alert properties.

To edit an alert by using a playbook, you must have one of the following XDR roles: Main administrator, SOC administrator, Tier 1 analyst, Tier 2 analyst, or Tenant administrator.

You cannot edit alerts that have the **Closed** status.

You can edit the following alerts properties by using the playbook:

- Assignee
- Alert status
- Comment
- ExternalReference attribute
- Additional data attribute

Below are example expressions you can enter in the **Show code** section of the visual playbook editor to change alert properties.

- **Assigning an alert to a user**

```
{
        "action": {
            "function": {
                "type": "assignAlert",
                    "params": {
                       "assignee": {
                          "id": "user_ID"
                        }
                    }
                }
            }
        }
```

When you edit an assignee in the playbook algorithm, suggestions are displayed. For convenience, the suggestions contain a search string where you can search by name. If you want to specify an alert assignee, you can search the corresponding record by the user's name, and the ID will be specified in the algorithm.

- **Assigning an alert to a user group**

```
{
    "action": {
      "function": {
         "type": "assignAlert",
          "params": {
            "group": {
               "id": "GROUP_ID"
                                }
                            }
                        }
                    }
          }
```

When you edit an assignee in the playbook algorithm, suggestions are displayed. For convenience, the suggestions contain a search string where you can search by the name of a user group. If you want to specify an alert assignee, you can search the corresponding record by the user's name, and the ID will be specified in the algorithm.

- **Unassigning an alert from a user**

```
{
            "action": {
                "function": {
                    "type": "assignAlert",
                    "params": {
                        "assignee": {
                            "id": "nobody"
                        }
                    }
                }
            }
        }
```

- **Unassigning an alert from a user group**

```
{
    "action": {
      "function": {
         "type": "assignAlert",
          "params": {
            "group": {
               "id": "nobody"
                                }
                        }
                    }
                }
            }
```

- **Changing the alert status**

To change the alert status to **In progress**:

```
{
            "action": {
                "function": {
                    "type": "setAlertStatus",
                        "params": {
                            "status": "inProgress"
                     }
                }
            }
        }
```

To change the alert status to **Closed**:

```
{
            "action": {
                "function": {
                    "type": "setAlertStatus",
                        "params": {
                            "status": "closed",
                            "statusResolution": "truePositive"
                     }
                }
```

```
            }
        }
```

You can also specify the following values for the statusResolution parameter: `falsePositive` and `lowPriority`.

When you edit an alert status in the playbook algorithm, the following suggestions can be displayed: `new`, `inProgress`, `closed`.

- **Adding a comment to an alert**

```
{
            "action": {
                "function": {
                    "type": "addCommentToAlert",
                    "params": {
                        "text": "${ \"New comment for alert with ID: \" + alert.InternalID }"
                    }
                }
            }
        }
```

- **Editing the ExternalReference attribute**

```
{
      "action": {
          "function": {
                "type": "setAlertExternalRef",
                   "params": {
                    "externalRef": "${ \"Appended externalRef for alert with ID: \" + alert.InternalID
}",
                      "mode": "append"
                    }
                }
            }
        }
```

To replace the current value of the ExternalReference attribute in the alert with the value from the playbook, specify the `replace` value for the `mode` parameter.

- **Editing the Additional data attribute**

```
{
      "action": {
           "function": {
              "type": "addAlertAdditionalData",
                 "params": {
                    "data": "${ {\"customKey_1 (alert.InternalID)\": (\"customValue_1 (\" +
alert.InternalID + \")\" )} }",
                    "mode": "append"
                    }
                }
```

```
        }
    }
```

To replace the current value of the AdditionalData attribute in the alert with the value from the playbook, specify the `replace` value for the `mode` parameter.

Managing Sandbox

Sandbox is a security component that utilizes virtual machines (VMs) with operating system images to analyze the behavior of objects, such as files, programs, or scripts, for potential malicious activity and signs of targeted attacks to the corporate IT infrastructure.

For Sandbox to process objects, you need to configure it, and then install virtual machines in it.

Initial Sandbox setup

Before installing virtual machines, you need to configure Sandbox:

- Configure database updates.
- Ensure that internet access was configured correctly during Sandbox installation. Adjust settings if necessary.

Optionally, you can change the administrator password and configure authentication policies.

Configuring database updates for Sandbox

During the license validity period, Sandbox obtains update packages automatically once every hour.

To configure database updates for Sandbox:

1. In the Sandbox web interface, go to **Database update**.
2. In the **Update source** settings group, select a source from which you want to receive updates:
   - **Kaspersky update server**

     Sandbox connects to the Kaspersky update server over HTTP and downloads up-to-date databases.
   - **Kaspersky update server (secure connection)**

     Sandbox connects to the Kaspersky update server over HTTPS to download up-to-date databases. It is recommended to use HTTPS for secure updates.
   - **Custom server**

     Sandbox connects to your FTP or HTTP server or to the folder with program databases on your computer to download up-to-date databases.
3. If you select **Custom server**, enter the full path to the folder containing the update package in the field under the setting name.

4. Decide whether to use a proxy server when downloading updates:

   - To enable the proxy server, turn on the toggle switch next to the **Proxy server** settings group name, and then specify the settings to connect to the proxy server:

     - Proxy server address and port number.

     - Proxy server user name and password.

   - To disable the proxy server, turn off the toggle switch next to the **Proxy server** settings group name.

5. Click **Apply** to save the settings.

Sandbox network interfaces

Configuring Sandbox network interfaces includes setting up DNS servers, configuring management network interfaces, managing internet access for processed objects, and adding static network routes.

Configuring DNS settings

To configure DNS:

1. In the Sandbox web interface, go to **Network interfaces**.

2. In the **Host name** field, specify the name of the server to install Sandbox. The name of the server must be in the FQDN format.

3. Click the **Add** button to the right of **DNS servers**.

4. In the empty field that appears, specify the IP address of the primary DNS server in IPv4 format.

5. Click the ✓ button to the right of the specified IP address.

The DNS server is added.

To add more DNS servers, repeat steps 2 through 5.

To remove a previously added DNS server, click 🗑 to the right of the specified DNS server IP address. Note that you can only remove additional DNS servers, not the primary DNS server. If you have added two and more DNS servers, you can remove any of them. The remaining DNS server becomes the primary server.

Configuring a management network interface

A management network interface is intended for providing access to the Sandbox server via the SSH protocol. Sandbox also receives objects via this interface.

Management network interface is initially configured during Sandbox installation. After Sandbox is installed, you can change the management network interface settings from the Sandbox web interface as described below.

To configure a management network interface:

1. In the Sandbox web interface, go to **Network interfaces**.
2. In the **Management interface** settings group from the **Interface** drop-down list, select a network interface that you want to use as a management interface.
3. Configure the management interface parameters:
   - **IP**: the IP address that you want to assign to this network interface if no IP address is assigned.
   - **Mask**: the mask of the network in which you want to use this network interface.
4. Click **Apply**.

The management network interface is configured.

Configuring Internet access for processed objects

Objects processed by Sandbox may attempt to act on the Internet via the network interface configured for Internet access. In this case, Sandbox can analyze the behavior of these objects on the Internet. If you block Internet access, the Sandbox analyzes the behavior of objects excluding their actions on the Internet.

Security risks and considerations

If your organization's security policy blocks internet access from local network computers, and the Sandbox network interface for internet access is open, the following scenario becomes possible:

1. A hacker adds a malicious application to a random file and then sends it to Sandbox for scanning from a local network computer.
2. The file is transferred outside the local network during the scan.

If Internet access for processed object is blocked, this risk is eliminated. However, this also means the reduced quality of alerts, because processed objects cannot be properly analyzed.

To configure the network interface used for Internet access of processed objects:

1. In the Sandbox web interface, go to **Network interfaces**.
2. In the **Internet interface** settings group from the **Interface** list, select a network interface that you want to use for Internet access of processed objects.
3. Configure the interface parameters:
   - **IP**: the IP address that you want to assign to this network interface.
   - **Mask**: the network mask in which you want to use this network interface.
   - **Default gateway**: gateway address of the network in which you want to use this network interface.
4. Click **Apply**.

The network interface used for Internet access of processed objects is configured.

Adding a static network route

To add a static network route:

1. In the Sandbox web interface, go to **Network interfaces**.
2. In the **Static routes** settings group, click the **Add** button.

   A set of empty fields appears in the list of static network routes.
3. Configure the route parameters:
   - **IP**: the IP address of the server for which you want to configure a static network route.
   - **Mask**: the subnet mask.
   - **Gateway**: the IP address of the gateway.
4. From the **Interface** list, select a network interface for which you want to add a static network route.
5. Click the ✓ button to the right of the specified route.
6. Click **Apply**.

The static network route is added.

After adding a static network route, you can further manage it by clicking the button to delete it or the button to modify it. When modifying, you will need to specify the same parameters as when creating a new route as described above.

Changing the Sandbox administrator account password

To change the Sandbox administrator account password:

1. In the Sandbox web interface, go to **Administration** → **Change password**.
2. In the **Current password** field, enter the password you have used to log in into Sandbox.
3. In the **New password** and **Confirm password** fields, enter a new password.
4. Click **Change password**.

Sandbox authentication policies

You can configure Sandbox authentication policies for Kaspersky Next XDR Expert users.

Termination of inactive sessions

If the user is not doing anything (the session is idle), we recommend terminating such a session. This reduces the risk of unauthorized access to Sandbox in cases when the user is actually done working with the application, but did not end the session.

Notifications about failed authentication attempts

Frequent failed attempts to log in to the web interface may indicate that a cybercriminal is trying to gain access to Sandbox by brute-forcing the password. To prevent such unauthorized access, we recommend enabling notifications about the failed authentication attempt limit being exceeded.

Terminating inactive Sandbox sessions

If the user is not doing anything (the session is idle), we recommend terminating such a session. This reduces the risk of unauthorized access to Sandbox in cases when the user is actually done working with the application, but did not end the session.

To enable and configure the termination of inactive sessions:

1. In the Sandbox web interface, go to **Authentication policies**.
2. In the **Terminate inactive sessions** section, specify the time after which you want the user session to be terminated if the user does not perform any actions. Enter the time in minutes in the **Terminate inactive sessions after** field.
3. Click **Apply**.

Sandbox terminates inactive user sessions after the specified time.

Notifications about failed Sandbox authentication attempts

Frequent failed attempts to log in to the web interface may indicate that a cybercriminal is trying to gain access to Sandbox by brute-forcing the password. To prevent such unauthorized access, we recommend enabling notifications about the failed authentication attempt limit being exceeded.

To enable and configure notifications about failed authentication attempts:

1. In the Sandbox web interface, go to **Authentication policies**.
2. In the **Count authentication attempts** section set the **Count attempts** toggle switch to **Enabled**.
3. In the **Counting period** field, enter the time period for which you want to count failed authentication attempts. The time is specified in minutes. If a user attempts to log in to the web interface or the management console and fails more than two times in a row, the following takes place:
   - A record about the user's failed authentication attempts is written to the user activity log.
   - If a rule has been created for sending notifications about too many failed authentication attempts, an email message is sent to the addresses specified in the notification rule. The message contains information about the failed user authentication attempts.

In the **Counting period** field, you may specify `0`, in which case all failed authentication attempts are counted, regardless of the how much time has passed between them.

4. Click **Apply**.

Setting up sending detected objects to the KPSN denylist

The application allows users with roles Tier 1 analyst and Tier 2 analyst to enable or disable automatic sending of malicious file hashes scanned by the Sandbox to the KPSN denylist. The configuration of this parameter is individual for each tenant. It can be changed only in the root tenant of the OSMP installation: in child tenants, the option cannot be changed.

Before enabling automatic sending of detected objects to the KPSN denylist, make sure the KPSN usage is enabled and the connection to a local reputation database of KPSN is configured.

To enable automatic sending of detected objects to the KPSN denylist:

1. In the main menu, go to **Settings** → **Tenants**.
2. Click the **Root tenant** and go to the **Settings** → **KPSN reputation database** subsection.
3. Turn on the **Assign the "Untrusted" status to objects** toggle button.
4. Click **Save**.

Making virtual machines ready for work

Sandbox uses virtual machines to run objects on various operating systems in order to analyze the behavior of the objects and detect their malicious activity.

You can install virtual machines of two types:

- *Preconfigured*, or *default*. These are virtual machines based on operating system images (ISO files) from the distribution kit. You cannot modify these images or the applications installed there.

  The default operating systems and applications may require activation. Activation is performed using your licenses.

- *Custom*. These are virtual machines based on operating systems you choose. You can install any applications on these operating systems.

You can use custom and default operating system images at the same time.

Installing virtual machines involves the following steps:

1. *Optional*. Uploading images to Storage. You may skip this step and upload an image directly from your computer when creating a template.
2. Making a template by either importing a preconfigured template or creating a custom virtual machine template.

3. *For custom templates*. Configuring the operating system and software.

4. Creating a virtual machine.

5. Installing a virtual machine.

Uploading images to Storage

To upload an image to Storage:

1. In the Sandbox web interface, go to **Templates & Storage** → **Storage**.

2. Click **Upload**.

3. Select the .ISO file that you want to upload to Storage, and then click **Open**.

Importing a template

Importing a template is the first step if you want to install a preconfigured virtual machine.

You can also import a template of a custom virtual machine if you have created and exported it earlier.

To import a virtual machine template:

1. In the Sandbox web interface, go to **Templates & Storage** → **Templates**.

2. From the **Add** drop-down list, select **Import template**.

3. Select a file to import, and then click **Open**.

Creating a custom virtual machine template

To create a custom virtual machine template:

1. In the Sandbox web interface, go to **Templates & Storage** → **Templates**.

2. From the **Add** drop-down list, select **Create template**.
   Template creation window opens.

3. At the **Preparing the image** step:

   a. Specify image **Name** and optional **Description**.

   b. To specify an operating system for the template, do one of the following:

      - Select an image from the list of available images in the **OS image** drop-down list.

      - Upload the operating system image: click **Upload**, select an ISO-file, and then click **Open**.

   c. Click **Proceed**.

4. At the **Customizing the template** step:

   a. In the **Mount ISO** drop-down list, select the image of the application that you want to install in the operating system.

      For the image to be displayed in the list, do one of the following:

      - Upload the image to Storage.

      - Upload the application image file: click **Upload**, select an ISO-file, and then click **Open**.

   b. When the application is installed, unmount the installed image: in the **Mount ISO** drop-down list, click the ⏏ icon next to the image.

   c. Configure the operating system and installed software.

Configuring the operating system and software

Preparing operating systems for use

When installing Windows 7, 8.1, or 10 operating systems, you must satisfy the following requirements:

- Disable the screen saver.

- Select the Always On power plan.

- Disable automatic updates.

- Disable Windows Firewall.

If you are using Windows 7, support of the SHA-2 hash algorithm is required. To make sure this hash algorithm is supported, install the Security Update for Windows 7 for x64-based Systems (KB3033929). For 32-bit Windows 7 operating systems, update KB3033929 must also be installed.

> Do not install update KB4474419. This update may cause a crash during virtual machine deployment.

To use Windows 7, you must enable TLS 1.1 and TLS 1.2 in the operating system settings. To do this, in Windows 7, in the **Control Panel** → **Internet Options** → **Advanced** section, select the **Use TLS 1.1** and **Use TLS 1.2** check boxes.

When using Windows 8.1 and 10 operating systems, you must disable fast startup and enable autologon.

Localization of operating systems

You can select a localization language when installing the system or install the system with English localization and additionally install a localization package in that system. You can select one of the following languages: Russian, English, Chinese (Simplified), Arabic, Spanish (Mexico). If you select a different localization, the quality of object scanning is diminished.

> For the Windows XP operating system, only Russian and English are supported.

Configuring operating systems

When the operating system is installed:

- Make sure the default command shell is configured.
- Activate the operating system and other licensed software.

You can do the following with the installed operating system:

- Assign a static name to the computer.
- Create user accounts.

  > In this case, you need to configure automatic logon.

- Install software.

  Limitations on software installation:

  - Only one image at a time can be connected to one template. After the template has been saved, you can disconnect one image and mount another.
  - Versions of Microsoft Office later than 2016 are not supported.
  - Installing the following types of software is strongly discouraged:
    - Software that injects its code into another running process
    - Drivers for protection
    - Anti-virus applications including Windows Defender
  - Detection of malicious activity of files that rely on highly specialized software to run is not guaranteed.

Kaspersky Next XDR Expert does not notify about problems with software installed on the operating system.

Creating a virtual machine

To create a virtual machine with a custom operating system image, internet access is required.

To create a virtual machine:

1. *For custom templates*. Make sure that the template used to create a virtual machine is turned off:

   a. In the Sandbox web interface, go to **Templates & Storage** → **Templates**.

   b. Select the template.

   c. If the template is turned on, select one of the options in the **Shut down** drop-down list:

   - **Shut down** if you want to shut down the system while saving the results of running applications.
   - **Power off** if you want to shut down the system without saving the results.

2. In the Sandbox web interface, go to **Virtual machines**.
3. Click **Create VM**.
4. In the **Template** drop-down list, select a template.
5. *For custom operating systems*: Specify the virtual machine **Name** and optional **Description**.
6. Click **Save**.
7. *For the preconfigured operating systems*: Some operating systems from the distribution kit require accepting the terms of end user license agreement. Read the text of the end user license agreement and click **Accept**.
8. *For custom operating systems*: If internet access is not configured for the server on which you are creating the virtual machine, the *No internet access* error message is displayed in the **Templates** window. To complete the virtual machine creation process, you must download debug symbols.

Installing a virtual machine

To install a virtual machine with a preconfigured operating system:

1. In the Sandbox web interface, go to **Virtual machines** → **Preconfigured**.
2. In the **Not installed virtual machines** section, click **Install ready VMs**.

To install a virtual machine with a custom operating system:

1. In the Sandbox web interface, go to **Virtual machines** → **Custom**.
2. Find the virtual machine in the list, and then click **Install** in the **Actions** column.
3. When the virtual machine is installed, in the **Actions** column, click **Enable**.

Downloading debug symbols

If internet access is not configured for the server on which the virtual machine with the custom image is installed, you must download the Microsoft debug symbols to correctly complete the virtual machine installation.

> For debug symbols to download correctly, the operating system used for the virtual machine template must have Windows Debug Tools installed and the name of the host connected to the network (hostname) must contain only Latin letters, numerals, and special characters.

To download debug symbols:

1. In the Sandbox web interface, go to **Templates & Storage** → **Templates**.
2. Select the template.
3. In the window with the **No internet access** error, click **Download manifesto**.

   > **Download manifesto** is not available until the virtual machine gets the **Failed** status.

4. Unpack the downloaded archive.
5. Run sbsymtool.ps1 using Windows PowerShell.
   This script downloads an archive with the debug symbols.
6. In the Sandbox web interface, go to **Templates & Storage** → **Templates**.
7. Select the template for which you have downloaded debug symbols.
8. In the **Actions** drop-down list, select **Upload symbols**.
9. In the window that opens, select the archive with the debug symbols, and then click **Open**.

Debug symbols are downloaded.

Configuring OSMP to use detection technologies

Kaspersky Next XDR Expert detects threat by using different detection technologies such as Anti-Malware Engine or Sandbox. You need to configure OSMP to be able to use the detection technologies.

Anti-Malware Engine

For Anti-Malware Engine, do the following:

- Configure getting updates for anti-virus databases.
- Create a list of passwords for password-protected archives.

Sandbox

After you have installed and configured Sandbox and installed virtual machines in it, you need to do the following:

- Configure getting updates for anti-virus databases.
- Select operating systems for scanning objects.
- Create custom Sandbox rules if you are going to use custom operating systems.
- Configure sending files from Endpoint Agents.

YARA

For YARA scanning, do the following:

Import YARA rules

IOC

For IOC scanning, do the following:

- Configure IOC scanning on schedule
- Import IOC rules

Updating anti-virus databases for Sandbox and Anti-Malware Engine

When scanning objects, Sandbox and Anti-Malware Engine require up-to-date anti-virus databases, so the anti-virus databases should be updated regularly. As a source of updates, you can use one of Kaspersky server or your own source, which can be an HTTP server, FTP server, or folder.

Kaspersky Next XDR Expert automatically looks for new update packages in the specified source once every 30 minutes. You can also update the anti-virus databases manually at any time.

> To configure getting updates for Sandbox and Anti-Malware Engine, you must have one of the following XDR roles: Main administrator, SOC administrator, Tenant administrator, Auditor.
> Users with the Auditor role can only view the current update source. Users with the other role can view and change the update source.

To configure getting updates for Sandbox and Anti-Malware Engine:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the **Root** tenant (update settings cannot be changed in other tenants).

   The tenant's properties window opens.

3. Select the **Settings** tab.

4. In the **Scanning** section, select the **Update databases** tab.

5. In the **Update source** drop-down list, select one of update sources:

   - **Kaspersky update servers (secure connection)**

     Kaspersky Next XDR Expert connects to Kaspersky update server over HTTPS and downloads up-to-date databases. It is recommended to use HTTPS to download database updates.

   - **Kaspersky update servers**

     Kaspersky Next XDR Expert connects to Kaspersky update server over HTTP and downloads up-to-date databases.

   - **Custom server**

     Kaspersky Next XDR Expert connects to your update source to download up-to-date databases. Specify the path to your update source in **URL or folder path**. It can be either a URL for HTTP or FTP server, or a full path to the folder containing an update package.

6. If you want to update the anti-virus databases immediately, click **Update**.

7. Click **Save**.

Getting updates for Sandbox and Anti-Malware Engine is configured. Kaspersky Next XDR Expert displays information about the latest update in the **Updated on** field.

You can update the anti-virus databases manually at any time by clicking **Update**.

Creating a list of passwords for password-protected archives

By default, Kaspersky Next XDR Expert cannot scan objects in password-protected archives. Such archives must be unpacked first. You can create a list of passwords commonly used for archives in your organization. Kaspersky Next XDR Expert tries passwords from the list when scanning an archive. If one of the passwords matches, Kaspersky Next XDR Expert unpacks the archive and scans objects in it.

To work with the list of passwords for password-protected archives, you must have one of the following XDR roles: Main administrator, Tenant administrator, Tier 1 analyst, Tier 2 analyst, Auditor.
Users with the Auditor role can only view the list of passwords. Users with the other role can view and edit the list.

To create a list of passwords for unpacking password-protected archives:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the **Root** tenant (the list of passwords cannot be changed in other tenants).

   The tenant's properties window opens.

3. Select the **Settings** tab.

4. In the **Scanning** section, select the **Archives** tab.

5. In the **Password for archives** field, add passwords for unpacking password-protected archives, each password on a new line.

6. If you want to restrict access to the list of passwords, turn on the **Require a password to access the password list** toggle switch, and then specify the password in the **Set access password** pane. You can share this password with the OSMP users who need access to the list of passwords for password-protected archives.

   Click **Save** to save the access password.

   You can change this password later if necessary by clicking **Change password** on the **Archives** tab.

7. Click **Save** or **Save and close** to save the list of passwords for password-protected archives.

Selecting operating systems for scanning objects in Sandbox

The Sandbox component runs objects in several operating systems and analyzes the behavior of these objects to detect malicious activity and indicators of targeted attacks and intrusions into the corporate IT infrastructure.

You can choose operating systems for scanning objects. It can be one of predefined sets, a custom set or their combination.

To select a set of operating systems:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Select the **Settings** tab.

4. In the **Sandbox servers** section, click the link next to **Operating systems to check**.

5. In the **Edit operating system set** pane that opens, under **Set of operating systems**, select either one of the predefined sets of operating systems or **Custom**.

   If you want to combine preset operating systems with custom operating systems, select **Custom**.

6. If you have selected **Custom**, the list of available operating systems is displayed under **Set contents**.

   This list always includes preset operating systems. If there is no virtual machine running a preset operating system on Sandbox, this operating system has the **Unknown** status.

   The list includes a custom operating system if there is a virtual machine running this operating system on the Sandbox server.

   Select the check boxes next to the operating systems that you want to include in the set.

7. Click **Save**.

The set of operating systems for scanning objects is ready.

For a custom set of operating systems, you also need to create custom rules.

Creating custom Sandbox rules

If you have selected custom operating systems for scanning objects, you also need to create custom rules that define what kind of objects should be scanned and on which operating system.

You do not need to create rules for sending objects for scanning to the preset operating systems, Kaspersky Next XDR Expert sends objects there automatically.

> Custom Sandbox rules apply only to custom virtual machines created in the Sandbox web interface.

To add a custom Sandbox rule:

1. In the main menu, go to **Monitoring** → **Resources and services** → **Sandbox rules**.

2. Click **Add**.

   Rule creation window opens.

3. Configure the following settings:

   - Turn on the **Rule status** toggle switch if you want to enable the rule immediately after it is created.
   - Select the tenant owner.
   - Select the virtual machine on which files should be scanned.

     You can only choose from virtual machines running custom operating systems that you have selected.
   - Under **Matching criteria**, select whether you want to use a **Mask**, a **File category** or **Both** as a matching criteria.
   - If you have selected **Mask** or **Both** as a matching criteria:
     - Specify a **Mask** for the names of files to be sent to Sandbox according to the rule.

       You can specify multiple values, each value on its own line.

       You can use the * and ? wildcard characters. Other wildcard characters are not supported.
     - Specify an optional **Mask exclusion** for files that must be excluded from scanning even if their file extensions match the mask specified earlier.

       You can specify multiple values, each value on its own line.

       You can use the * and ? wildcard characters. Other wildcard characters are not supported.

     > You can use the **Mask exclusion** field if only virtual machines tagged as **Custom** are selected in the **Custom set** list of the **Sandbox servers** settings. If both a custom virtual machine and a preconfigured virtual machine are selected, the **Mask exclusion** field is ignored.
   - If you have selected **File category** or **Both** as a matching criteria:

     In the **File category** drop down list, select categories of files to be sent to Sandbox according to the rule.
   - To limit the size of files to be sent to Sandbox, click **Add file size**, and then specify the **Minimum** and **Maximum** size of a file. This way you can set multiple size ranges.

4. Click **Save**.

The rule appears in the list of custom Sandbox rules.

Specifics of scanning archives

Archives are scanned as follows:

1. Kaspersky Next XDR Expert unpacks the archive.
2. If some files from the archive match the rule, Kaspersky Next XDR Expert sends them for scanning.

Files with the MSI extension are processed similar to archives.

Configuring sending of files from Endpoint Agents to Sandbox

Endpoint Agents can send files to Sandbox for scanning in several ways:

- Endpoint Agent **automatically** detects objects and submits them to Sandbox for scanning.
- Endpoint Agent user **manually** submits a file for scanning.
- Endpoint Agent sends a file that has triggered an **IOA rule**. When an IOA rule is triggered on a device, and this IOA rule contains a condition for sending detected objects to Sandbox, Kaspersky Next XDR Expert automatically initiates the **Get file** response with the **Scan file** option enabled.

Automatic sending of files to Sandbox may significantly increase the load on OSMP.

To configure sending of files to be scanned in Sandbox:

1. Make sure Sandbox is integrated with OSMP.
2. In the main menu, go to **Settings** → **Tenants**.
   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.
3. Click the name of the required tenant.
   The tenant's properties window opens.
4. Select the **Settings** tab.
5. In the **Scanning** section, select the **Sandbox** tab.
6. Configure the settings:
   - To enable **automatic** sending of files detected by Endpoint Agent:
     Under **Send files from devices for analysis to Sandbox**, turn on the **Send files automatically** toggle switch.
     By default, this feature is disabled.
   - To enable **manual** sending of files by a user:
     Under **Send files from devices for analysis to Sandbox**, turn on the **Send files manually** toggle switch.
     By default, this feature is enabled.
   - To enable sending of files when an **IOA rule** is triggered:
     Under **Send files to Sandbox automatically**, turn on the **Send files** toggle switch.
     By default, this feature is disabled.
7. Click **Save and close**.

Sending of files to Sandbox is configured. Based on the results of the scan, Kaspersky Next XDR Expert may create an alert.

About indicators of compromise, IOC files and IOC rules

Kaspersky Next XDR Expert can scan devices for indicators of compromise. Indicators of compromise (IOCs) are digital objects like file hashes or IP addresses that might signal a cyberattack. You can learn more about IOCs in Kaspersky IT encyclopedia .and on Securelist .

IOC files

Multiple IOCs can be combined into a structured description in an *IOC file*. There is a standard *OpenIOC format* for such files.

Below is an example of a simple IOC file.

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="unique
ID in the UUID format" last-modified="date of the last modification of the rule in the YYYY-MM-DDThh:mm:ss format"
xmlns="http://schemas.mandiant.com/2010/ioc">
        <short_description>Rule name</short_description>
        <authored_by>Rule name</authored_by>
        <authored_date>file creation date in the YYYY-MM-DDThh:mm:ss format</authored_date>
        <links />
        <definition>
                <Indicator operator="OR" id="Unique ID in the UUID format">
                        <IndicatorItem id="Unique ID in the UUID format" condition="is">
                                <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
                                <Content type="md5">MD5 hash of the file</Content>
                        </IndicatorItem>
                </Indicator>
        </definition>
</ioc>
```

You can create your own IOC file in a text editor and save it with the .ioc extension. Depending on your Endpoint Agent application, you can use OpenIOC terms listed in files:

OpenIOC terms for Kaspersky Endpoint Security for Windows

OpenIOC terms for Kaspersky Endpoint Security for Linux

There is no OpenIOC terms list for Kaspersky Endpoint Security for Mac, because Kaspersky Next XDR Expert does not send IOC scan tasks to Kaspersky Endpoint Security for Mac.

IOC rules

When you import an IOC rule, Kaspersky Next XDR Expert extracts the structured description from an IOC file and creates a rule based on the description.

IOC rules are available in the IOC rules list in the OSMP Console interface → **Monitoring** → **Resources and services** → **IOC rules**.

Importing IOC rules

Kaspersky Next XDR Expert imports IOC rules from files containing IOC rules. One file can contain only one IOC rule, so the list of IOC rules is also the list of IOC files from which the rules were created.

The list of IOC rules is available in the OSMP Console interface → **Monitoring** → **Resources and services** → **IOC rules**.

To import an IOC rule:

1. In the main menu, go to **Monitoring** → **Resources and services** → **IOC rules**.

   The list of IOC rules opens.

2. Click **Import**.

3. In the **Import IOC rule** pane that opens, configure the following settings:

   - Select the **Tenant**.

   - Select the **Importance** of the rule. Kaspersky Next XDR Expert assigns this value to an alert created as the result of the IOC rule being triggered.

   - If you want Kaspersky Next XDR Expert to regularly scan devices for indicators of compromise described in the rule, turn on the **Scheduled scan** toggle switch (the scheduled scan must be configured).

     You can also enable scheduled scan later by turning on a toggle switch in the **Scheduled scan** column of the IOC rules list.

   - Specify the description (optional).

4. **Upload a file** containing an IOC rule.

5. Click **Import**.

Imported rule appears in the list of IOC rules.

Importing YARA rules

Kaspersky Next XDR Expert uses YARA rules to scan files sent to OSMP and to scan devices.

Kaspersky Next XDR Expert imports YARA rules from a file containing YARA rules. One file can contain multiple YARA rules. Kaspersky Next XDR Expert parses the file, extracts the rules and adds them to the list of rules.

The list of YARA rules is available in the OSMP Console interface → **Monitoring** → **Resources and services** → **YARA rules**.

To import YARA rules:

1. In the main menu, go to **Monitoring** → **Resources and services** → **YARA rules**.

   The list of YARA rules opens.

2. Click **Import**.

3. In the **Import YARA rules** pane that opens, configure the following settings:

   - Select the **Tenant**.

   - The **Importance** of any YARA rule is **High** by default, you cannot change it. Kaspersky Next XDR Expert assigns this value to an alert created as the result of the YARA rule being triggered.

   - If you want Kaspersky Next XDR Expert to use all imported rules to scan files sent to OSMP, turn on the **File scanning** toggle switch.

     You can also include any rule to the set of rules for file scanning later by turning on a toggle switch in the **File scanning** column of the YARA rules list.

     > This feature is only available for the Root tenant.

   - Specify the description (optional).

4. **Upload a file** containing YARA rules.

   When the file is uploaded, Kaspersky Next XDR Expert extracts rules from the file and shows information about the extracted rules.

5. Click **Import**.

Imported rules appear in the list of YARA rules.

REST API

You can access XDR from third-party solutions using the API. The XDR REST API operates over HTTP and consists of a set of request/response methods.

REST API Help Swagger: XDR

REST API Help Swagger: KUMA

On the Swagger page, use the **Select a definition** drop-down list to switch between KUMA and XDR (OSMP) API sets.

REST API requests must be sent to the following address:

https://api.<XDR FQDN>/xdr/api/v1/<request>

https://api.<XDR FQDN>/xdr/api/v3/kuma/<request> (for KUMA-specific API)

Example:

https://api.example.com/xdr/api/v1/

https://api.example.com/xdr/api/v3/kuma/ (for KUMA-specific API)

Creating a token

To generate a user API token:

1. In the main menu, go to **Settings** → **API Tokens**.
2. Click **Add token**.
3. In the **Add token** panel, configure the token options:
   a. Click **Expiration date** and use the calendar to specify the expiration date. If you want to disable automatic expiration for the token, select the **No expiration date** check box.

      The maximum expiration date range is 365 days.

      > We recommend that you enable automatic expiration for tokens that have access to POST methods.

   b. Select check boxes next to the API methods you want to allow access to.
4. Click **Generate**.

5. Click **Copy and close**.

> You will not be able to copy the token later.

The token is created and copied to the clipboard. Save the token in any convenient way.

Authorizing API requests

Each API request must include token-based authorization. The user whose token is used to make the API request must have the permissions to perform this type of request.

Each request must be accompanied by the following header:

```
Authorization: Bearer <token>
```

Possible errors

HTTP code	Description	`message` field value
400	Invalid header	invalid authorization header
403	The token does not exist or the owner user is disabled	access denied

Working with Open Single Management Platform

Open Single Management Platform (hereinafter referred to as OSMP) is an open technology platform that allows you to integrate Kaspersky applications and third-party applications into a single security system, and provide cross-application scenarios. Kaspersky Next XDR Expert is based on OSMP. To manage Kaspersky Next XDR Expert, the OSMP web interface (hereinafter referred to as OSMP Console) is used.

Using OSMP Console, you can do the following:

- Manage the status of the organization's security system.
- View information about the security of your organization's network.
- Configure the detection, hunting, and response of threats.
- Manage policies created for assets on your network.
- Manage tasks for applications installed on your network devices.
- Manage users and roles.
- Configure the migration of data to Kaspersky Next XDR Expert.
- Install Kaspersky applications on devices on your network and manage installed applications.
- Poll the network to discover client devices, and distribute the devices to administration groups manually or automatically.
- Manage Kaspersky Next XDR Expert integrations with other applications.

OSMP Console is a multi-language web interface. You can change the interface language at any time, without reopening the application.

The default authentication method in OSMP Console is to enter user name, password, and an additional single-use security code for two-factor authentication. Starting with Kaspersky Next XDR Expert 2.0, two-factor authentication is enabled automatically and the option to configure two-factor authentication at sign-in is disabled by default.

You can configure domain authentication over the Kerberos protocol. Domain authentication allows you to enable secure authentication in OSMP Console without having to re-enter the password on the corporate network (single sign-on).

Configuring Administration Server

This section describes the configuration process and properties of Kaspersky Security Center Administration Server.

Configuring the Administration Server connection address

The Administration Server connection address is the network address used for the following purposes:

- As a default address when creating the Network Agent installation packages

  After the installation of the Network Agent, managed devices connect to the Administration Server at this address.

- As the KSN proxy server address

  Managed devices connect to the KSN proxy server via this address.

To configure the Administration Server connection address:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **General** section.

3. In the window that opens, in the **Administration Server connection address** field, specify the new Administration Server connection address.

   The maximum length of the connection address is 255 characters.

   You can specify the Administration Server connection address as the NetBIOS name, FQDN, or the IP address of the Administration Server.

4. Click the **Save** button.

The specified address is used as the Administration Server address and the KSN proxy server address.

Configuring the connection of OSMP Console to Administration Server

You can configure the connection of OSMP Console to Administration Server through the Administration Server properties or by using the Administration Server policy settings.

To set the connection ports through the Administration Server properties:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Connection ports** section.

3. If necessary, specify the **SSL port for Web Console** setting or specify other Administration Server connection ports.

The main connection settings of the selected Server are specified.

To set the connection ports through the Administration Server policy settings:

1. In the main menu, go to **Asset management** → **Policies**.

2. Click the name of the Administration Server policy, and then go to the **Application settings** tab.

3. If necessary, specify the **SSL port for Web Console** setting or specify other Administration Server connection ports.

   If you disable the **Open port for Web Console** option and this policy setting will be applied to the device, you will not be able to connect to the Administration Server by using OSMP Console. In this case, the connection will be terminated. If you have an Administration Server on which this policy is not applied, you can reconnect to this Administration Server through OSMP Console.

The main connection settings of the selected Server are specified.

Configuring internet access settings

An internet connection is required for proper operation of Kaspersky Next XDR Expert components and can be used for specific integrations, both Kaspersky and third-party. For example, you must configure internet access for Administration Server to use Kaspersky Security Network and to download updates of anti-virus databases for Open Single Management Platform and managed Kaspersky applications.

The integration settings of some Kaspersky applications contain an option to enable or disable the usage of proxy server. For example, such an option is available when you configure integration with Kaspersky Threat Intelligence Portal.

To specify the internet access settings:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Configuring internet access** section.

3. Enable the **Use proxy server** option if you want to use a proxy server when connecting to the internet. If this option is enabled, the fields are available for entering settings. Specify the following settings for a proxy server connection:

   - **Address**

     Address of the proxy server used for Open Single Management Platform connection to the internet.

   - **Port number**

     Number of the port through which Open Single Management Platform proxy connection will be established.

- **Bypass proxy server for local addresses**

  No proxy server will be used to connect to devices in the local network.

- **Proxy server authentication**

  If this check box is selected, in the entry fields you can specify the credentials for proxy server authentication.

  This entry field is available if the **Use proxy server** check box is selected.

- **User name**

  User account under which connection to the proxy server is established (this field is available if the **Proxy server authentication** check box is selected).

- **Password**

  Password set by the user under whose account the proxy server connection is established (this field is available if the **Proxy server authentication** check box is selected).

  To see the entered password, click and hold the **Show** button for as long as you require.

Certificates for work with Open Single Management Platform

This section contains information about Open Single Management Platform certificates and describes how to issue and replace certificates for OSMP Console and how to renew a certificate for Administration Server if the Server interacts with OSMP Console.

About Open Single Management Platform certificates

Open Single Management Platform uses the following types of certificates to enable a secure interaction between the application components:

- Administration Server certificate
- OSMP Console Server certificate
- OSMP Console certificate

By default, Open Single Management Platform uses self-signed certificates (that is, issued by Open Single Management Platform itself), but you can replace them with custom certificates to better meet the requirements of your organization's network and comply with the security standards. After Administration Server verifies whether a custom certificate meets all applicable requirements, this certificate assumes the same functional scope as a self-signed certificate. The only difference is that a custom certificate is not reissued automatically upon expiration. You replace certificates with custom ones by means of the klsetsrvcert utility or through the Administration Server properties section in OSMP Console, depending on the certificate type. When you use the klsetsrvcert utility, you need to specify a certificate type by using one of the following values:

- C—Common certificate for ports 13000 and 13291.
- CR—Common reserve certificate for ports 13000 and 13291.

The maximum validity period for any of the Administration Server certificates must be 397 days or less.

Administration Server certificates

An Administration Server certificate is required for the following purposes:

- Authentication of Administration Server when connecting to OSMP Console
- Secure interaction between Administration Server and Network Agent on managed devices
- Authentication when the primary Administration Servers are connected to secondary Administration Servers

The Administration Server certificate is created automatically during installation of the Administration Server component and it is stored in the /var/opt/kaspersky/klnagent_srv/1093/cert/ folder. You specify the Administration Server certificate when you create a response file to install OSMP Console. This certificate is called common ("C").

The Administration Server certificate is valid for 397 days. Open Single Management Platform automatically generates a common reserve ("CR") certificate 90 days before the expiration of the common certificate. The common reserve certificate is subsequently used for seamless replacement of the Administration Server certificate. When the common certificate is about to expire, the common reserve certificate is used to maintain the connection with Network Agent instances installed on managed devices. With this purpose, the common reserve certificate automatically becomes the new common certificate 24 hours before the old common certificate expires.

> The maximum validity period for any of the Administration Server certificates must be 397 days or less.

If necessary, you can assign a custom certificate for the Administration Server. For example, this may be necessary for better integration with the existing PKI of your enterprise or for custom configuration of the certificate fields. When replacing the certificate, all Network Agents that were previously connected to Administration Server through SSL will lose their connection and will return "*Administration Server authentication error.*" To eliminate this error, you will have to restore the connection after the certificate replacement.

If the Administration Server certificate is lost, you must reinstall the Administration Server component, and then restore the data in order to recover it.

You can also back up the Administration Server certificate separately from other Administration Server settings in order to move Administration Server from one device to another without data loss.

Mobile certificates

A mobile certificate ("M") is required for authentication of the Administration Server on mobile devices. You specify the mobile certificate in the Administration Server properties.

Also, a mobile reserve ("MR") certificate exists: it is used for seamless replacement of the mobile certificate. Open Single Management Platform automatically generates this certificate 60 days before the expiration of the common certificate. When the mobile certificate is about to expire, the mobile reserve certificate is used to maintain the connection with Network Agent instances installed on managed mobile devices. With this purpose, the mobile reserve certificate automatically becomes the new mobile certificate 24 hours before the old mobile certificate expires.

If the connection scenario requires the use of a client certificate on mobile devices (connection involving two-way SSL authentication), you can generate those certificates by means of the certificate authority for auto-generated user certificates ("MCA"). Also, in the Administration Server properties, you can specify custom client certificates issued by a different certification authority, while integration with the domain Public Key Infrastructure (PKI) of your organization enables you to issue client certificates by means of your domain certification authority.

Web Server certificate

Web Server, a component of Kaspersky Security Center Administration Server, uses a special type of certificate. This certificate is required for publishing Network Agent installation packages that you subsequently download to managed devices. For this purpose, Web Server can use various certificates.

Web Server uses one of the following certificates, in order of priority:

1. Custom Web Server certificate that you specified manually by means of OSMP Console

2. Common Administration Server certificate ("C")

OSMP Console certificate

The OSMP Console Server has its own certificate. When you open a website, a browser verifies whether your connection is trusted. The Web Console certificate allows you to authenticate the Web Console and is used to encrypt traffic between a browser and the Web Console.

When you open the Web Console, the browser may inform you that the connection to the OSMP Console is not private and the OSMP Console certificate is invalid. This warning appears because the OSMP Console certificate is self-signed and automatically generated by Open Single Management Platform. To remove this warning, you can do one of the following:

- Replace the Web Console certificate with a custom one (recommended option). Create a certificate that is trusted in your infrastructure and that meets the requirements for custom certificates.

- Add the Web Console certificate to the list of trusted browser certificates. We recommend that you use this option only if you cannot create a custom certificate.

Requirements for custom certificates used in Open Single Management Platform

The table below shows the requirements for custom certificates specified for different components of Open Single Management Platform.

Requirements for Open Single Management Platform certificates

Certificate type	Requirements	Comments
Common certificate, Common reserve certificate ("C", "CR")	Minimum key length: 2048. Basic constraints: • Path Length Constraint: None Key Usage: • Digital signature • Certificate signing • Key encryption • CRL Signing Extended Key Usage (optional): server authentication, client authentication. The certificate must include a valid Subject Alternative Name (SAN) that contains the Administration Server loopback address: • For IPv4: 127.0.0.1 • For IPv6: 0:0:0:0:0:0:0:1	Extended Key Usage parameter is optional. Path Length Constraint value may be an integer different from "None", but not less than 1.
Mobile certificate, Mobile reserve certificate ("M", "MR")	Minimum key length: 2048. Basic constraints: • CA: true • Path Length Constraint: None Key Usage: • Digital signature • Certificate signing • Key encryption • CRL Signing Extended Key Usage (optional): server authentication.	Extended Key Usage parameter is optional. Path Length Constraint value may be an integer different from "None", if Common certificate has a Path Length Constraint value not less than 1.
Certificate CA for auto-generated user certificates ("MCA")	Minimum key length: 2048. Basic constraints: • CA: true • Path Length Constraint: None Key Usage: • Digital signature • Certificate signing • Key encryption • CRL Signing Extended Key Usage (optional): server authentication, client authentication.	Extended Key Usage parameter is optional. Path Length Constraint value may be an integer different from "None," if Common certificate has a Path Length Constraint value not less than 1.
Web Server certificate	Extended Key Usage: server authentication. The PKCS #12 / PEM container from which the certificate is specified includes the entire chain of public keys. The Subject Alternative Name (SAN) of the certificate is present; that is, the value of the `subjectAltName` field is valid. The certificate meets the effective requirements of browsers imposed on server certificates, as well as the current baseline requirements of the CA/Browser Forum.	—

Certificate type	Requirements	Comments
Kaspersky Security Center Web Console certificate	The PEM container from which the certificate is specified includes the entire chain of public keys. The Subject Alternative Name (SAN) of the certificate is present; that is, the value of the `subjectAltName` field is valid. The certificate meets the effective requirements of browsers to server certificates, as well as the current baseline requirements of the CA/Browser Forum.	Encrypted certificates are not supported by Kaspersky Security Center Web Console.
Intermediate certificate for public Kaspersky Next XDR Expert services	The PEM container that contains the complete certificate chain (or only one certificate) and an unencrypted private key. The intermediate certificate must be issued by your organization's private key infrastructure (PKI). The validity period for the custom intermediate certificate must be 399 days or more.	—

Reissuing the certificate for OSMP Console

Most browsers impose a limit on the validity term of a certificate. To fall within this limit, the validity term of the OSMP Console certificate is limited to 397 days. You can replace an existing certificate received from a certification authority (CA) by issuing a new self-signed certificate manually. Alternatively, you can reissue your expired OSMP Console certificate.

Automatically reissuing the certificate for OSMP Console is not supported. You have to manually reissue the expired certificate.

When you open the OSMP Console, the browser may inform you that the connection to the OSMP Console is not private and the OSMP Console certificate is invalid. This warning appears because the OSMP Console certificate is self-signed and automatically generated by Open Single Management Platform. To remove or prevent this warning, you can do one of the following:

- Specify a custom certificate when you reissue it (recommended option). Create a certificate that is trusted in your infrastructure and that meets the requirements for custom certificates.
- Add the OSMP Console certificate to the list of trusted browser certificates after you reissue the certificate. We recommend that you use this option only if you cannot create a custom certificate.

To reissue the expired OSMP Console certificate:

Reinstall OSMP Console by performing one of the following:

- If you want to use the same installation file of OSMP Console, remove OSMP Console, and then install the same OSMP Console version.
- If you want to use an installation file of an upgraded version, run the upgrade command.

The OSMP Console certificate is reissued for another validity term of 397 days.

Replacing certificate for OSMP Console

By default, when you install OSMP Console Server (also referred to as OSMP Console), a browser certificate for the application is generated automatically. You can replace the automatically generated certificate with a custom one.

To replace the certificate for OSMP Console with a custom one:

1. Open the response file that you created when installing the OSMP Console.

   If you want, you can create a new response file.

2. In the file, specify the following paths:

   - To the OSMP Console custom certificate file by using the `certPath` parameter
   - To the private key associated with the OSMP Console custom certificate by using the `keyPath` parameter.

   > You have to specify both the paths. Otherwise, the web browser keeps informing you that your connection is not private.

3. Reinstall OSMP Console by specifying the new response file. Do one of the following:

   - If you want to use the same installation file of OSMP Console, remove OSMP Console, and then install the same OSMP Console version.
   - If you want to use an installation file of an upgraded version, run the upgrade command.

OSMP Console works with the specified certificate.

Converting a PFX certificate to the PEM format

To use a PFX certificate in OSMP Console, you must first convert it to the PEM format by using any convenient OpenSSL-based cross-platform utility.

To convert a PFX certificate to the PEM format in the Linux operating system:

1. In an OpenSSL-based cross-platform utility, execute the following commands:

   ```
   openssl pkcs12 -in <filename.pfx> -clcerts -nokeys | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > server.crt
   ```

   ```
   openssl pkcs12 -in <filename.pfx> -nocerts -nodes | sed -ne '/-BEGIN PRIVATE KEY-/,/-END PRIVATE KEY-/p' > key.pem
   ```

2. Make sure that the certificate file and the private key are generated to the same directory where the .pfx file is stored.

3. OSMP Console does not support passphrase-protected certificates. Therefore, run the following command in an OpenSSL-based cross-platform utility to remove a passphrase from the .pem file:

   ```
   openssl rsa -in key.pem -out key-without-passphrase.pem
   ```

Do not use the same name for the input and output .pem files.

As a result, the new .pem file is unencrypted. You do not have to enter a passphrase to use it.

The .crt and .pem files are ready to use, so you can specify them in the OSMP Console installer.

Scenario: Specifying the custom Administration Server certificate

You can assign the custom Administration Server certificate, for example, for better integration with the existing public key infrastructure (PKI) of your enterprise or for custom configuration of the certificate fields. It is useful to replace the certificate immediately after installation of Administration Server.

The maximum validity period for any of the Administration Server certificates must be 397 days or less.

Prerequisites

The new certificate must be created in the PKCS#12 format (for example, by means of the organization's PKI) and must be issued by trusted certification authority (CA). Also, the new certificate must include the entire chain of trust and a private key, which must be stored in the file with the pfx or p12 extension. For the new certificate, the requirements listed below must be met.

Certificate type: Common certificate, common reserve certificate ("C", "CR")

Requirements:

- Minimum key length: 2048
- Basic constraints:
  - CA: true
  - Path Length Constraint: None
    Path Length Constraint value may be an integer different from "None" but not less than 1.
- Key Usage:
  - Digital signature
  - Certificate signing
  - Key encipherment
  - CRL Signing
- Extended Key Usage (EKU): server authentication and client authentication. The EKU is optional, but if your certificate contains it, the server and client authentication data must be specified in the EKU.
- The certificate must include a valid Subject Alternative Name (SAN).

- The `subjectAltName` field must contain the Administration Server loopback address:
  - For IPv4: 127.0.0.1
  - For IPv6: 0:0:0:0:0:0:0:1

> Certificates issued by a public CA do not have the certificate signing permission. To use such certificates, make sure that you installed Network Agent version 13 or later on distribution points or connection gateways in your network. Otherwise, you will not be able to use certificates without the signing permission.

Stages

Specifying the Administration Server certificate proceeds in stages:

1. **Replacing the Administration Server certificate**

   Use the command-line klsetsrvcert utility for this purpose.

2. **Specifying a new certificate and restoring connection of Network Agents to the Administration Server**

   When the certificate is replaced, all Network Agents that were previously connected to Administration Server through SSL lose their connection and return "Administration Server authentication error." To specify the new certificate and restore the connection, use the command-line klmover utility.

Results

When you finish the scenario, the Administration Server certificate is replaced and the server is authenticated by Network Agents on the managed devices.

Replacing the Administration Server certificate by using the klsetsrvcert utility

To replace the Administration Server certificate,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke ksc --action klsetsrvcert \
  --param ksc_server_certificate="<path_to_new_certificate>" \
  --param ksc_server_cert_pass="<password>"
```

where:

- `< path_to_new_certificate >` is the path to the container with the certificate and a private key in the PKCS #12 format (file with the .P12 or .PFX extension).
- `< password >` is the password used for protection of the PKCS #12 container. The certificate and a private key are stored in the container, therefore, the password is required to decrypt the file with the container.

  If the password contains double quotes, escape them with backslashes (`\"`).

By default, certificate validation parameters are not specified, a custom certificate without signing permission is used. You can replace the common certificate for port 13000.

> You do not need to download the klsetsrvcert utility. It is included in the Kubernetes cluster and is not available for direct running. You can run the klsetsrvcert utility only by using KDT from the administrator host.

Connecting Network Agents to Administration Server by using the klmover utility

After you replace the Administration Server certificate by using the command-line klsetsrvcert utility, you need to establish the SSL connection between Network Agents and Administration Server because the connection is broken.

To specify the new Administration Server certificate and restore the connection:

From the command line, run the following utility:

```
klmover [-address <server address>] [-pn <port number>] [-ps <SSL port number>] [-nossl] [-cert <path to certificate file>]
```

This utility is automatically copied to the Network Agent installation folder, when Network Agent is installed on a client device.

The description of the klmover utility parameters is presented in the table below.

Values of the klmover utility parameters

Parameter	Value
`-address < server address >`	Address of the Administration Server for connection. You can specify an IP address or the DNS name.
`-pn < port number >`	Number of the port through which non-encrypted connection to the Administration Server is established. The default port number is 14000.
`-ps < SSL port number >`	Number of the SSL port through which encrypted connection to the Administration Server is established by using SSL. The default port number is 13000. For the root Administration Server, this port is 13000 and it cannot be changed.
`-nossl`	Use non-encrypted connection to the Administration Server. If the key is not in use, Network Agent is connected to the Administration Server by using encrypted SSL protocol.
`-cert < path to certificate file >`	Use the specified certificate file for authentication of access to Administration Server.

Hierarchy of Administration Servers

Some client companies, for example MSP, may run multiple Administration Servers. It can be inconvenient to administer several separate Administration Servers, so a hierarchy can be applied. Each Administration Server can have several secondary Administration Servers on different nesting levels of the hierarchy. The root Administration Server can only act as a primary Server.

In a hierarchy, a Linux-based Administration Server can work both as a primary Server and as a secondary Server. The primary Linux-based Server can manage both Linux-based and Windows-based secondary Servers. A primary Windows-based Server can manage a secondary Linux-based Server.

A "primary/secondary" configuration for two Administration Servers provides the following options:

- A secondary Administration Server inherits policies, tasks, user roles, and installation packages from the primary Administration Server, thus preventing duplication of settings.
- Selections of devices on the primary Administration Server can include devices from secondary Administration Servers.
- Reports and event selections on the primary Administration Server can contain data (including detailed information) from secondary Administration Servers.
- A primary Administration Server can be used as a source of updates for a secondary Administration Server.

The primary Administration Server only receives data from non-virtual secondary Administration Servers within the scope of the options listed above. This limitation does not apply to virtual Administration Servers, which share the database with their primary Administration Server.

Creating a hierarchy of Administration Servers: adding a secondary Administration Server

In a hierarchy, a Linux-based Administration Server can work both as a primary Server and as a secondary Server. The primary Linux-based Server can manage both Linux-based and Windows-based secondary Servers. A primary Windows-based Server can manage a secondary Linux-based Server. The root Administration Server can only act as a primary Server.

You can add an Administration Server as a secondary Administration Server, thus establishing a "primary/secondary" hierarchy.

To add a secondary Administration Server that is available for connection through OSMP Console:

1. Make sure that port 13000 of the future primary Administration Server is available for receipt of connections from secondary Administration Servers.
2. Go to the properties of the future primary Administration Server:
   a. In the main menu, select any section (except for the **Marketplace**, **Console navigation**, and **Settings**), and then at the top of the page, click the **Show hierarchy** button.
   b. In the navigation pane that opens, hover the mouse cursor over the name of the required Administration Server, and then click the settings icon (⚙).

      Alternately, you can click the Administration Server name, and then, after the name is displayed in the main menu, click it in the main menu to open the server properties window.
3. In the properties window that opens, click the **Administration Servers** tab.
4. Select the check box next to the name of the administration group to which you want to add the Administration Server.

5. In the menu line, click **Connect secondary Administration Server**.

   The Add secondary Administration Server wizard starts. Proceed through the wizard by using the **Next** button.

6. Fill in the following fields:

   - **Secondary Administration Server display name**

     A name by which the secondary Administration Server will be displayed in the hierarchy. If you want, you can enter the IP address as a name, or you can use a name like, for example, "Secondary Server for group 1".

   - **Secondary Administration Server address (optional)**

     Specify the IP address or the domain name of the secondary Administration Server.

     This parameter is required if the **Connect primary Administration Server to secondary Administration Server in DMZ** option is enabled.

   - **Administration Server SSL port**

     Specify the number of the SSL port on the primary Administration Server. The default port number is 13000.

     For the root Administration Server, this port is 13000 and it cannot be changed.

   - **Administration Server API port**

     Specify the number of the port on the primary Administration Server for receiving connections over OpenAPI. The default port number is 13299.

   - **Connect primary Administration Server to secondary Administration Server in DMZ**

     Select this option if the secondary Administration Server is in a demilitarized zone (DMZ).

     If this option is selected, the primary Administration Server initiates connection to the secondary Administration Server. Otherwise, the secondary Administration Server initiates connection to the primary Administration Server.

7. Specify the certificate of the future secondary Server.

   The wizard is complete.

8. Send the certificate file of the future primary Administration Server to the system administrator of the office where the future secondary Administration Server is located. (You can, for example, write the file to an external device, such as a flash drive, or send it by email.)

   The certificate file is located on the future primary Administration Server, at /var/opt/kaspersky/klnagent_srv/1093/cert/.

9. Prompt the system administrator in charge of the future secondary Administration Server to do the following:

   a. Click the Administration Server name.

   b. On the properties page that opens, proceed to the **Hierarchy of Administration Servers** section of the **General** tab.

   c. Select the **This Administration Server is secondary in the hierarchy** option.

   > The root Administration Server can only act as a primary Server.

   d. In the **Primary Administration Server address** field, enter the network name of the future primary Administration Server.

   e. Select the previously saved file with the certificate of the future primary Administration Server by clicking **Browse**.

   f. If necessary, select the **Connect primary Administration Server to secondary Administration Server in DMZ** check box.

   g. If the connection to the future primary Administration Server is performed through a proxy server, select the **Use proxy server** option and specify the connection settings.

   h. Click **Save**.

The "primary/secondary" hierarchy is built. The primary Administration Server starts receiving connection from the secondary Administration Server using port 13000. The tasks and policies from the primary Administration Server are received and applied. The secondary Administration Server is displayed on the primary Administration Server, in the administration group where it was added.

Viewing the list of secondary Administration Servers

To view the list of the secondary (including virtual) Administration Servers:

1. In the main menu, select a section (except for **Marketplace**, **Console navigation**, and **Settings**).

2. At the top of the page, click the **Show hierarchy** button.

In the navigation pane that opens, the list of the secondary (including virtual) Administration Servers is displayed. You can proceed to any of these Administration Servers by clicking its name.

> You cannot proceed to virtual Administration Servers if you opened the navigation pane from the **IP ranges** and **Deleted objects** sections.
> The administration groups are shown, too. The groups are available for management if you opened the navigation pane from the following sections: **Assets (Devices)**, **Policies**, **Tasks**, and **Distribution points**. Otherwise, the groups are grayed-out.

If you are connected to your primary Administration Server in OSMP Console, and can not connect to a virtual Administration Server that is managed by a secondary Administration Server, you can use one of the following ways:

- **Modify the existing OSMP Console installation to add the secondary Server to the list of trusted Administration Servers**. Then you will be able to connect to the virtual Administration Server in OSMP Console.

  1. On the device where OSMP Console is installed, run the OSMP Console installation file corresponding to the Linux distribution installed on your device under an account with administrative privileges.

     The Setup Wizard will start. Proceed through the wizard by using the **Next** button.

  2. Select the **Upgrade** option.

  3. On the **Modification type** step, select the **Edit connection settings** option.

  4. On the **Trusted Administration Servers** step, add the required secondary Administration Server.

  5. On the last step, click **Modify** to apply the new settings.

  6. After the application reconfiguration successfully completes, click the **Finish** button.

- Use OSMP Console to connect directly to the secondary Administration Server where the virtual Server was created. Then you will be able to switch to the virtual Administration Server in OSMP Console.

Managing virtual Administration Servers

This section describes the following actions to manage virtual Administration Servers:

- Create virtual Administration Servers
- Enable and disable virtual Administration Servers
- Assign an administrator for a virtual Administration Server
- Change the Administration Server for client devices
- Delete virtual Administration Servers

Creating a virtual Administration Server

You can create virtual Administration Servers and add them to administration groups.

When you create a virtual Administration Server, it inherits the user list and all of the user rights of the primary Administration Server. If a user has access rights to the primary Server, this user has access rights to the virtual Server as well. After creation, you configure the access rights to the Servers independently.

To create and add a virtual Administration Server:

1. In the main menu, click the Administration Server name.

2. On the page that opens, proceed to the **Administration Servers** tab.

3. Select the administration group to which you want to add a virtual Administration Server.
   The virtual Administration Server will manage devices from the selected group (including the subgroups).

4. On the menu line, click **New virtual Administration Server**.

5. On the page that opens, define the properties of the new virtual Administration Server:

   - **Name of virtual Administration Server**.

   - **Administration Server connection address**

     You can specify the name or the IP address of your Administration Server.

6. From the list of users, select the virtual Administration Server administrator. If you want, you can edit one of the existing accounts before assigning it the administrator's role, or create a new user account.

   If necessary, you can create a new account that will act as a virtual Administration Server administrator by using the kladduser utility. To do that, use the following command:

   ```
   kladduser -n <user_name> -p <password> -vs <virtual_server_name>
   ```

   where `virtual_server_name` is the name of the virtual Administration Server.

7. Click **Save**.

The new virtual Administration Server is created, added to the administration group and displayed on the **Administration Servers** tab.

If you are connected to your primary Administration Server in OSMP Console, and can not connect to a virtual Administration Server that is managed by a secondary Administration Server, you can use one of the following ways:

- **Modify the existing OSMP Console installation to add the secondary Server to the list of trusted Administration Servers**. Then you will be able to connect to the virtual Administration Server in OSMP Console.

  1. On the device where OSMP Console is installed, run the OSMP Console installation file corresponding to the Linux distribution installed on your device under an account with administrative privileges.

     The Setup Wizard will start. Proceed through the wizard by using the **Next** button.

  2. Select the **Upgrade** option.

  3. On the **Modification type** step, select the **Edit connection settings** option.

  4. On the **Trusted Administration Servers** step, add the required secondary Administration Server.

5. On the last step, click **Modify** to apply the new settings.

6. After the application reconfiguration successfully completes, click the **Finish** button.

- Use OSMP Console to connect directly to the secondary Administration Server where the virtual Server was created. Then you will be able to switch to the virtual Administration Server in OSMP Console.

Enabling and disabling a virtual Administration Server

When you create a new virtual Administration Server, it is enabled by default. You can disable or enable it again at any time. Disabling or enabling a virtual Administration Server is equal to switching off or on a physical Administration Server.

To enable or disable a virtual Administration Server:

1. In the main menu, click the Administration Server name.

2. On the page that opens, proceed to the **Administration Servers** tab.

3. Select the virtual Administration Server that you want to enable or disable.

4. On the menu line, click the **Enable / disable virtual Administration Server** button.

   The virtual Administration Server state is changed to enabled or disabled, depending on its previous state. The updated state is displayed next to the Administration Server name.

Assigning an administrator for a virtual Administration Server

When you use virtual Administration Servers in your organization, you might want to assign a dedicated administrator for each virtual Administration Server. For example, this might be useful when you create virtual Administration Servers to manage separate offices or departments of your organization, or if you are an MSP provider and you manage your tenants through virtual Administration Servers.

When you create a virtual Administration Server, it inherits the user list and all of the user rights of the primary Administration Server. If a user has access rights to the primary Server, this user has access rights to the virtual Server as well. After creation, you configure the access rights to the Servers independently. If you want to assign an administrator for a virtual Administration Server only, make sure that the administrator does not have access rights on the primary Administration Server.

You assign an administrator for a virtual Administration Server by granting the administrator access rights to the virtual Administration Server. You can grant the required access rights in one of the following ways:

- Configure access rights for the administrator manually

- Assign one or more user roles for the administrator

To sign in to OSMP Console, an administrator of a virtual Administration Server specifies the virtual Administration Server name, user name, and password. OSMP Console authenticates the administrator and opens the virtual Administration Server to which the administrator has access rights. The administrator cannot switch between Administration Servers.

Prerequisites

Before you start, ensure that the following conditions are met:

- The virtual Administration Server is created.
- On the primary Administration Server, you have created an account for the administrator that you want to assign for the virtual Administration Server.
- You have the **Modify object ACLs** right in the **General features** → **User permissions** functional area.

Configuring access rights manually

To assign an administrator for a virtual Administration Server:

1. Switch to the required virtual Administration Server:

   a. In the main menu, select a section (except for **Marketplace**, **Console navigation**, and **Settings**), and then at the top of the page, click the **Show hierarchy** button.

   b. In the navigation pane that opens, click the name of the required virtual Administration Server.

   > You cannot switch to a virtual Administration Server from the **IP ranges** and **Deleted objects** sections.

   In the main menu, the Administration Server changes to the one that you selected.

2. Move to the properties of the administration Server by clicking its name in the main menu.

   Alternately, in the navigation pane, hover the mouse cursor over the name of the required virtual Administration Server, and then click the settings icon (⚙).

   The Administration Server properties window opens.

3. On the **Access rights** tab, click the **Add** button.

   A unified list of users of the primary Administration Server and the current virtual Administration Server opens.

4. From the list of users, select the account of the administrator that you want to assign for the virtual Administration Server, and then click the **OK** button.

   The application adds the selected user to the user list on the **Access rights** tab.

5. Select the check box next to the added account, and then click the **Access rights** button.

6. Configure the rights that the administrator will have on the virtual Administration Server.

   For successful authentication, at minimum, the administrator must have the following rights:

   - **Read** right in the **General features** → **Basic functionality** functional area
   - **Read** right in the **General features** → **Virtual Administration Servers** functional area

The application saves the modified user rights to the administrator account.

Configuring access rights by assigning user roles

Alternatively, you can grant the access rights to a virtual Administration Server administrator through user roles. For example, this might be useful if you want to assign several administrators on the same virtual Administration Server. If this is the case, you can assign the administrators' accounts the same one or more user roles instead of configuring the same user rights for several administrators.

To assign an administrator for a virtual Administration Server by assigning user roles:

1. On the primary Administration Server, create a new user role, and then specify all of the required access rights that an administrator must have on the virtual Administration Server. You can create several roles, for example, if you want to separate access to different functional areas.

2. Switch to the required virtual Administration Server:

   a. In the main menu, select a section (except for **Marketplace**, **Console navigation**, and **Settings**), and then at the top of the page, click the **Show hierarchy** button.

   b. In the navigation pane that opens, click the name of the required virtual Administration Server.

   > You cannot switch to a virtual Administration Server from the **IP ranges** and **Deleted objects** sections.

   In the main menu, the Administration Server changes to the one that you selected.

3. Assign the new role or several roles to the administrator account.

The application assigns the roles to the administrator account.

Configuring access rights at the object level

In addition to assigning access rights at the functional area level, you can configure access to specific objects on the virtual Administration Server, for example, to a specific administration group or a task. To do this, switch to the virtual Administration Server, and then configure the access rights in the object's properties.

Changing the Administration Server for client devices

You can change the Administration Server that manages client devices to a different Server using the *Change Administration Server* task. After the task completion, the selected client devices will be put under the management of the Administration Server that you specify.

> You cannot use the *Change Administration Server* task for client devices connected to Administration Server through connection gateways. For such devices you have to either reconfigure Network Agent or reinstall Network Agent and specify connection gateway.

To change the Administration Server that manages client devices to a different Server:

1. In the main menu, go to **Asset management** → **Tasks**.

2. Click **Add**.

   The New task wizard starts. Proceed through the wizard by using the **Next** button.

3. At the **New task settings** step of the wizard, specify the following settings:

   a. In the **Application** drop-down list, select **Open Single Management Platform**.

   b. In the **Task type** field, select **Change Administration Server**.

   c. In the **Task name** field, specify the name for the task that you are creating.

      A task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).

   d. Select devices to which the task will be assigned:

      - **Assign task to an administration group**

        The task is assigned to devices included in an administration group. You can specify one of the existing groups or create a new one.

        For example, you may want to use this option to run a task of sending a message to users if the message is specific for devices included in a specific administration group.

        If a task is assigned to an administration group, the **Security** tab is not displayed in the task properties window because group tasks are subject to the security settings of the groups to which they apply.

      - **Specify device addresses manually or import addresses from a list**

        You can specify DNS names, IP addresses, and IP subnets of devices to which you want to assign the task.

        You may want to use this option to execute a task for a specific subnet. For example, you may want to install a certain application on devices of accountants or to scan devices in a subnet that is probably infected.

      - **Assign task to a device selection**

        The task is assigned to devices included in a device selection. You can specify one of the existing selections.

        For example, you may want to use this option to run a task on devices with a specific operating system version.

4. At the **Task scope** step of the wizard, specify an administration group, devices with specific addresses, or a device selection.

5. At this step of the wizard, confirm that you agree to the terms of changing the Administration Server for client devices.

6. At this step of the wizard, select the Administration Server that you want to use to manage the selected devices:

   - **Change to another primary Administration Server**

     To move client devices to another primary Administration Server, specify the following Administration Server connection settings:

     1. In the **Administration Server address** field, specify the address of the new primary Administration Server.
     2. In the **Port number** field, specify the port number to connect to Administration Server.

        The default port number is 14000.
     3. In the **SSL port** field, specify the number of the SSL port on the primary Administration Server.

        The default port number is 13000.
     4. If necessary, enable the **Use proxy server** option.

        If this option is disabled, direct connection is used to connect the device to the Administration Server.

        If this option is enabled, specify the proxy server parameters:

        - **Proxy server address**
        - **Proxy server port**

        If your proxy server requires authentication, in the **User name** and **Password** fields, specify the credentials of the account under which connection to the proxy server is established. We recommend that you specify the credentials of an account that has minimum privileges required only for the proxy server authentication.
     5. If necessary, upload a new Administration Server certificate.

   - **Change to another virtual Server on this primary Server**

     Select this option to move client devices to virtual Administration Server on the current primary Administration Server. To do this, in the **Name of virtual Administration Server** drop-down list, select the necessary virtual Administration Server.

7. At the **Selecting an account to run the task** step of the wizard, specify the account settings:

   - **Default account**

     The task will be run under the same account as the application that performs this task.

     By default, this option is selected.

- **Specify account**

  Fill in the **Account** and **Password** fields to specify the details of an account under which the task is run. The account must have sufficient rights for this task.

  - **Account**

    Account under which the task is run.

  - **Password**

    Password of the account under which the task will be run.

8. If you want to change the default task settings, at the **Finish task creation** step of the wizard, enable the **Open task details when creation is complete** option.

   If you do not enable this option, the task is created with the default settings. You can change the default settings later, at any time.

9. Click the **Finish** button.

   The task is created and displayed in the list of tasks.

10. Click the name of the created task to open the task properties window.

11. If you want to change the default task settings, in the task properties window, specify the general task settings according to your needs.

12. Click the **Save** button.

    The task is created and configured.

13. Run the created task.

After the task is completed, the client devices for which it was created are put under the management of the Administration Server specified in the task settings.

Deleting a virtual Administration Server

When you delete a virtual Administration Server, all of the objects created on the Administration Server, including policies and tasks, will be deleted as well. The managed devices from the administration groups that were managed by the virtual Administration Server will be removed from the administration groups. To return the devices under management of Kaspersky Next XDR Expert, run the network polling, and then move the found devices from the Unassigned devices group to the administration groups.

To delete a virtual Administration Server:

1. Switch to the primary physical Administration Server to which the virtual Administration Server to be deleted belongs:

   a. In the main menu, select any section (except for **Marketplace**, **Console navigation**, and **Settings**), and then at the top of the page, click the **Show hierarchy** button.

   b. In the navigation pane that opens, click the name of the primary physical Administration Server.

   In the main menu, the Administration Server changes to the one that you selected.

2. Move to the properties of the primary physical Administration Server in one of the following ways:

   - In the navigation pane, hover the mouse cursor over the name of the required Administration Server, and then click the settings icon (⚙).

   - In the main menu, click the Administration Server name.

3. On the properties window that opens, proceed to the **Administration Servers** tab.

4. Select the virtual Administration Server that you want to delete.

5. On the menu line, click the **Delete** button.

The virtual Administration Server is deleted.

Configuring Administration Server connection events logging

The history of connections and attempts to connect to the Administration Server during its operation can be saved to a log file. The information in the file allows you to track not only connections inside your network infrastructure, but unauthorized attempts to access the server as well.

To log events of connection to the Administration Server:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Connection ports** section.

3. Enable the **Log Administration Server connection events** option.

All further events of inbound connections to the Administration Server, authentication results, and SSL errors will be saved to the file /var/opt/kaspersky/klnagent_srv/logs/SC.syslog.

Setting the maximum number of events in the event repository

In the **Events repository** section of the Administration Server properties window, you can edit the settings of events storage in the Administration Server database by limiting the number of event records and record storage term. When you specify the maximum number of events, the application calculates an approximate amount of storage space required for the specified number. You can use this approximate calculation to evaluate whether you have enough free space on the disk to avoid database overflow. The default capacity of the Administration Server database is 400,000 events. The maximum recommended capacity of the database is 45 million events.

The application checks the database every 10 minutes. If the number of events reaches the specified maximum value plus 10,000, the application deletes the oldest events so that only the specified maximum number of events remains.

When the Administration Server deletes old events, it cannot save new events to the database. During this period, information about events that were rejected is written to the operating system log. The new events are queued and then saved to the database after the deletion operation is complete. By default, the event queue is limited to 20,000 events. You can customize the queue limit by editing the KLEVP_MAX_POSTPONED_CNT flag value.

To limit the number of events that can be stored in the events repository on the Administration Server:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Events repository** section. Specify the maximum number of events stored in the database.

3. Click the **Save** button.

Additionally, you can do the following:

- Change the settings of any task to save events related to the task progress, or save only task execution results.

- Reduce or disable the storage period for the events of Administration Server, Network Agent, and Kaspersky applications installed on managed devices.

In doing so, you will reduce the number of events in the database, increase the speed of execution of scenarios associated with analysis of the event table in the database, and lower the risk that critical events will be overwritten by a large number of events.

Changing DBMS credentials

Sometimes, you may need to change DBMS credentials, for example, in order to perform a credential rotation for security purposes.

Changing the Administration Server database credentials

To change Administration Server database credentials in a Linux environment by using the klsrvconfig utility:

1. Stop Administration Server. Run the following commands:

`systemctl stop kladminserver_srv.service`

`systemctl stop klnagent_srv.service`

`systemctl stop klactprx_srv.service`

`systemctl stop klwebsrv_srv.service`

2. Run the klsrvconfig utility:

   `sudo /opt/kaspersky/ksc64/sbin/klsrvconfig -set_dbms_cred`

3. Specify a new account name. You should specify credentials of an account that exists in the DBMS.

4. Enter a new password.

5. Specify the new password for confirmation.

6. Run Administration Server. Run the following commands:

   `systemctl start kladminserver_srv.service`

   `systemctl start klnagent_srv.service`

   `systemctl start klactprx_srv.service`

   `systemctl start klwebsrv_srv.service`

The Administration Server database credentials are changed.

Changing the IAM database credentials

To change the IAM database credentials in a Linux environment by modifying the configuration file:

1. Stop Administration Server. Run the following commands:

   `systemctl stop kladminserver_srv.service`

   `systemctl stop klnagent_srv.service`

   `systemctl stop klactprx_srv.service`

   `systemctl stop klwebsrv_srv.service`

2. Locate the IAM configuration file: /var/opt/kaspersky/klnagent_srv/iam/iam_config.yaml

3. In the `dbms_iam` section, specify new account credentials in the user and password fields. You should specify credentials of an account that exists in the DBMS.

4. Run Administration Server. Run the following commands:

   `systemctl start kladminserver_srv.service`

   `systemctl start klnagent_srv.service`

   `systemctl start klactprx_srv.service`

   `systemctl start klwebsrv_srv.service`

The IAM database credentials are changed.

Backup copying and restoration of the Administration Server data

Data backup allows you to save the Administration Server data in a certain state, and restore the data if needed, for example, if the Administration Server data is corrupted.

Before you back up the Administration Server data, check whether a virtual Administration Server is added to the administration group. If a virtual Administration Server is added, make sure that an administrator is assigned to this virtual Administration Server before the backup. You cannot grant the administrator access rights to the virtual Administration Server after the backup. Note that if the administrator account credentials are lost, you will not be able to assign a new administrator to the virtual Administrator Server.

You can create a backup copy of the Administration Server data only by running the *Backup of Administration Server data* task. This task is automatically created when you deploy Kaspersky Next XDR Expert.

On the primary Administration Server, creating and removing the *Backup of Administration Server data* task is not available.

The backup copy is saved in the `/var/spool/ksc/backup` directory. The backup directory is automatically created on the worker node on which Administration Server is installed when you deploy Kaspersky Next XDR Expert. On the primary Administration Server, you cannot change the backup directory path.

The following data is saved in the backup copy of Administration Server:

- Database of Administration Server (policies, tasks, application settings, events saved on the Administration Server)
- Configuration details of the structure of administration groups and client devices
- Repository of distribution packages of applications for remote installation
- Administration Server certificate

Recovery of the Administration Server data is only possible by using the KDT utility.

You can create a backup copy of the KUMA Core and restore it from the backup if needed. You can also back up other Kaspersky Next XDR Expert components by using third-party tools only if you use the DBMS installed on a separate server outside the Kubernetes cluster. You must not create the Administration Server database backup by using third-party tools.

Configuring the Administration Server Backup task

The *Administration Server Backup* task automatically is created when you deploy Kaspersky Next XDR Expert and cannot be deleted. You can create a backup copy of Administration Server data only by running the *Administration Server Backup* task.

To configure the Backup of Administration Server data task:

1. In the main menu, go to **Asset management** → **Tasks**, and then select the **Administration Server Backup** task.
2. Click the **Administration Server Backup** task.

   The task properties window opens.
3. If necessary, specify the general task settings according to your needs.
4. In the **Application settings** section, set the backup protection password and number of backup copies if needed.

   We recommend limiting the number of Administration Server data backups, to avoid overflow in the disk space allocated to store backups.
5. Click **Save** to apply changes.

The *Backup of Administration Server data* task is configured.

Using the KDT utility to recover Administration Server data

The *Backup of Administration Server data* task allows you to copy Administration Server data for backup. To recover Administration Server data, you must use the KDT utility.

To recover Administration Server data:

1. On the administrator host where the KDT utility is located, run the following command:

   `./kdt invoke ksc --action listBackup`

   The list of backups located in the `/var/spool/ksc/backup` directory is displayed.
2. Run the following command:

```
./kdt invoke ksc --action restoreBackup \
  --param ksc_file_backup="<file name>" \
  --param ksc_backup_password="<password>"
```

where:

- `ksc_file_backup` is the path to the required backup archive and the archive name.
- `ksc_backup_password` is the archive password if the backup was saved with a password. If no password was used set the `ksc_backup_password` variable to "".

  If the password contains double quotes, escape them with backslashes (`\"`).

The Administration Server data is recovered from the selected backup archive.

Deleting a hierarchy of Administration Servers

If you no longer want to have a hierarchy of Administration Servers, you can disconnect them from this hierarchy.

To delete a hierarchy of Administration Servers:

1. In the main menu, click the Administration Server name.
2. On the page that opens, proceed to the **Administration Servers** tab.
3. In the administration group from which you want to delete the secondary Administration Server, select the secondary Administration Server.
4. On the menu line, click **Delete**.
5. In the window that opens, click **OK** to confirm that you want to delete the secondary Administration Server.

The former primary Administration Server and the former secondary Administration Server are now independent of each other. The hierarchy no longer exists.

Access to public DNS servers

If access to Kaspersky servers by using the system DNS is not possible, Open Single Management Platform can use the following public DNS servers, in the following order:

1. Google Public DNS (8.8.8.8)
2. Cloudflare DNS (1.1.1.1)
3. Alibaba Cloud DNS (223.6.6.6)
4. Quad9 DNS (9.9.9.9)
5. CleanBrowsing (185.228.168.168)

Requests to these DNS servers may contain domain addresses and the public IP address of the Administration Server, because the application establishes a TCP/UDP connection to the DNS server. If Open Single Management Platform is using a public DNS server, data processing is governed by the privacy policy of the relevant service.

To configure the use of public DNS by using the klscflag utility:

1. On the administrator host where the KDT utility is located, run the following command to disable the use of public DNS:

   ```
   ./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv
   ".core/.independent" -s Transport -n ForceUseSystemDNS -t d -v 1'
   ```

2. To enable the use of public DNS, run the following command:

   ```
   ./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv
   ".core/.independent" -s Transport -n ForceUseSystemDNS -t d -v 0'
   ```

Configuring the interface

You can configure the OSMP Console interface to display and hide sections and interface elements, depending on the features being used.

To configure the OSMP Console interface in accordance with the currently used set of features:

1. In the main menu, go to **Settings** → **Interface options**.

2. Enable or disable the required options:

   - **Show data encryption and protection**

   - **Show EDR alerts**

3. Click **Save**.

   After the required options are enabled, the console displays the corresponding sections in the main menu. For example, if you enable **Show EDR alerts**, the **Monitoring** → **Alerts** section appears in the main menu.

Encrypt communication with TLS

To fix vulnerabilities on your organization's corporate network, you can enable traffic encryption by using the TLS protocol. You can enable TLS encryption protocols and supported cipher suites on Administration Server. Open Single Management Platform supports the TLS protocol versions 1.0, 1.1, 1.2, and 1.3. You can select the required encryption protocol and cipher suites.

Open Single Management Platform uses self-signed certificates. You can also use your own certificates. We recommend using certificates issued by trusted certificate authorities.

To configure allowed encryption protocols and cipher suites on Administration Server:

1. On the administrator host where the KDT utility is located, run the following command:

   ```
   ./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv
   ".core/.independent" -s Transport -n SrvUseStrictSslSettings -v <value> -t d'
   ```

   Use the SrvUseStrictSslSettings flag to configure allowed encryption protocols and cipher suites on Administration Server.

Specify the <value> parameter of the SrvUseStrictSslSettings flag:

- 4—Only the TLS 1.2 and TLS 1.3 protocols are enabled. Also, cipher suites with TLS_RSA_WITH_AES_256_GCM_SHA384 are enabled (these cipher suites are needed for backward compatibility with Kaspersky Security Center 11). This is the default value.

  Cipher suites supported for the TLS 1.2 protocol:

  - ECDHE-RSA-AES256-GCM-SHA384
  - ECDHE-RSA-AES256-SHA384
  - ECDHE-RSA-CHACHA20-POLY1305
  - AES256-GCM-SHA384 (cipher suite with TLS_RSA_WITH_AES_256_GCM_SHA384)
  - ECDHE-RSA-AES128-GCM-SHA256
  - ECDHE-RSA-AES128-SHA256

  Cipher suites supported for the TLS 1.3 protocol:

  - TLS_AES_256_GCM_SHA384
  - TLS_CHACHA20_POLY1305_SHA256
  - TLS_AES_128_GCM_SHA256
  - TLS_AES_128_CCM_SHA256

- 5—Only the TLS 1.2 and TLS 1.3 protocols are enabled. For the TLS 1.2 and TLS 1.3 protocols, the specific cipher suites listed below are supported.

  Cipher suites supported for the TLS 1.2 protocol:

  - ECDHE-RSA-AES256-GCM-SHA384
  - ECDHE-RSA-AES256-SHA384
  - ECDHE-RSA-CHACHA20-POLY1305
  - ECDHE-RSA-AES128-GCM-SHA256
  - ECDHE-RSA-AES128-SHA256

  Cipher suites supported for the TLS 1.3 protocol:

  - TLS_AES_256_GCM_SHA384
  - TLS_CHACHA20_POLY1305_SHA256
  - TLS_AES_128_GCM_SHA256
  - TLS_AES_128_CCM_SHA256

> We do not recommend using 0, 1, 2, or 3 as the parameter value of the SrvUseStrictSslSettings flag. These parameter values correspond to insecure TLS protocol versions (the TLS 1.0 and TLS 1.1 protocols) and insecure cipher suites, and are used only for backward compatibility with earlier Kaspersky Security Center versions.

2. Restart the following Open Single Management Platform services:
   - Administration Server
   - Web Server
   - Activation Proxy

Traffic encryption by using the TLS protocol is enabled.

You can use the KLTR_TLS12_ENABLED and KLTR_TLS13_ENABLED flags to enable the support of the TLS 1.2 and TLS 1.3 protocols, respectively. These flags are enabled by default.

To enable or disable the support of the TLS 1.2 and TLS 1.3 protocols,

On the administrator host where the KDT utility is located, run one of the following commands:

- To enable or disable the support of the TLS 1.2 protocol:

  ```
  ./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv
  ".core/.independent" -s Transport -n KLTR_TLS12_ENABLED -v <value> -t d'
  ```

- To enable or disable the support of the TLS 1.3 protocol:

  ```
  ./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv
  ".core/.independent" -s Transport -n KLTR_TLS13_ENABLED -v <value> -t d'
  ```

Specify the `<value>` parameter of the flag:

- `1`—To enable the support of the TLS protocol.
- `0`—To disable the support of the TLS protocol.

Backup copying and restoration of KUMA Core under Kaspersky Next XDR Expert

You can create a backup copy of KUMA Core and restore it from the backup copy, if needed.

Creating a backup copy of KUMA Core

To create a backup copy of KUMA Core, run one of the following commands:

- If you want to create a backup copy with the default name:

  ```
  ./kdt invoke kuma --action backup
  ```

  The default name of the backup copy contains the creation date and time, such as kuma-backup-YYYY-MM-DDTHH_MM_SS.tar.gz.

- If you want to create a backup copy with a custom name:

  ```
  ./kdt invoke kuma --action backup --of kuma_backup_output_file=./<custom name of kuma backup copy>-{{.Date}}.tar.gz
  ```

  By default, the backup copy is created in the current directory of administrator host.

Restoring KUMA Core from a backup copy

You may need to restore KUMA Core from a backup copy when KUMA Core is available, but some data is lost, services are missing, or are not working properly.

To restore KUMA Core from a backup copy:

1. Place the backup copy on the host where you plan to restore the backup.

2. Specify the backup copy name in the `kuma_backup_file` parameter, and then run the following command:

   ```
   ./kdt invoke kuma --action restore --if kuma_backup_file=./kuma-backup-YYYY-MM-DDTHH_MM_SS.tar.gz
   ```

3. If you need to force-clean the files from the /opt/kaspersky/kuma directory on KUMA Core before you restore it from the backup copy, specify the parameter `clean_restore=true explicitly`:

   ```
   ./kdt invoke kuma --action restore --if kuma_backup_file=./kuma-backup-YYYY-MM-DDTHH_MM_SS.tar.gz --param clean_restore=true
   ```

   KUMA Core is restored from a backup, along with the related data.

Reinstalling KUMA Core by using a backup copy

You may need to reinstall KUMA Core by using a backup copy created in the current deployment of Kaspersky Next XDR Expert in the following cases:

- KUMA Core is unavailable or corrupted.

- KUMA Core components in the cluster are corrupted or removed.

- The KUMA Core database is unavailable.

To install KUMA Core by using the backup file:

1. Run the following command:

   ```
   ./kdt apply -k kuma_<kuma version number>.tar --accept-eula -p --force --if kuma_backup_file=./kuma-backup-YYYY-MM-DDTHH_MM_SS.tar.gz
   ```

   > You cannot use a backup copy for KUMA standalone for this purpose.

As a result, the KUMA Core cluster components and executed files are restored, and files with current certificates for the services on host with external KUMA services are removed from the inventory file. The KUMA Core database is restored in the cluster.

2. After the command is executed, go to KUMA Console, select the **Active services** section, and then reset the certificates for all external services. The new certificates are issued and the services are connected to KUMA Core under Kaspersky Next XDR Expert.

KUMA Core is reinstalled by using the backup file.

Discovering networked devices

This section describes search and discovery of networked devices.

Open Single Management Platform allows you to find devices on the basis of specified criteria. You can save search results to a text file.

The search and discovery feature allows you to find the following devices:

- Managed devices in administration groups of Kaspersky Security Center Administration Server and its secondary Administration Servers.
- Unassigned devices managed by Kaspersky Security Center Administration Server and its secondary Administration Servers.

Scenario: Discovering networked devices

You must perform device discovery before installation of the security applications. When all networked devices are discovered, you can receive information about them and manage them through policies. Regular network polls are needed to discover if there are any new devices and whether previously discovered devices are still on the network.

Discovery of networked devices proceeds in stages:

1. **Initial device discovery**

   Perform device discovery manually.

2. **Configuring future polls**

   Make sure that IP range polling is enabled and that the poll schedule meets the needs of your organization. When configuring the poll schedule, use the recommendations for network polling frequency.

   You can also enable Zeroconf polling if your network includes IPv6 devices.

   If networked devices are included in a domain, it is recommended to use domain controller polling.

   > You can perform IP range polling and Zeroconf polling only by using a distribution point.

   Windows network polling is not supported.

3. **Setting up rules for adding discovered devices to administration groups (optional)**

   If new devices appear on your network, they are discovered during regular polls and are automatically included in the **Unassigned devices** group. If you want, you can set up the rules for automatically moving these devices to the **Managed devices** group. You can also establish retention rules.

   If you skip this rule-setting stage, all the newly discovered devices go to the **Unassigned devices** group and stay there. If you want, you can move these devices to the **Managed devices** group manually. If you move the devices to the **Managed devices** group manually, you can analyze information about each device and decide whether you want to move it to an administration group, and, if so, to which group.

Results

Completion of the scenario yields the following:

- Kaspersky Security Center Administration Server discovers the devices that are on the network and provides you with information about them.
- Future polls are set up and are conducted according to the specified schedule.

The newly discovered devices are arranged according to the configured rules. (Or, if no rules are configured, the devices stay in the **Unassigned devices** group).

Windows network polling

About Windows network polling

Open Single Management Platform Administration Server does not support Windows network polling.

During a quick poll, distribution points only retrieve information from the list of the NetBIOS names of devices in all network domains and workgroups. During a full poll, the following information is requested from each client device:

- Operating system name
- IP address
- DNS name
- NetBIOS name

Both quick polls and full polls require the following:

- Ports UDP 137/138, TCP 139, UDP 445, TCP 445 must be available in the network.
- The SMB protocol is enabled.
- The Microsoft Computer Browser service must be used, and the primary browser computer must be enabled on the Administration Server.

- The Microsoft Computer Browser service must be used, and the primary browser computer must be enabled on the client devices:
  - On at least one device, if the number of networked devices does not exceed 32.
  - On at least one device for each 32 networked devices.

The full poll can run only if the quick poll has run at least once.

IP range polling

Kaspersky Next XDR Expert allows you to poll an IP range only by using a distribution point. The distribution point attempts to perform reverse name resolution for every IPv4 address from the specified range to a DNS name, by using standard DNS requests. If this operation succeeds, the distribution point sends an `ICMP ECHO REQUEST` (the same as the `ping` command) to the received name. If the device responds, information about it is added to the Kaspersky Next XDR Expert database. The reverse name resolution is necessary to exclude network devices that can have an IP address but are not computers, for example, network printers or routers.

This polling method relies upon a correctly configured local DNS service. It must have a reverse lookup zone. If this zone is not configured, IP subnet polling will yield no results.

Initially, the distribution point gets IP ranges for polling from the network settings of the device assigned as a distribution point. If the device address is 192.168.0.1 and the subnet mask is 255.255.255.0, the network 192.168.0.0/24 is included in the list of polling address automatically. The distribution point polls all addresses from 192.168.0.1 to 192.168.0.254.

If only IP range polling is enabled, the distribution point discovers devices only with IPv4 addresses. If your network includes IPv6 devices, turn on Zeroconf polling of devices.

For all Linux-based hosts within a Microsoft Active Directory or Samba domain controller, specify the `netbios name` and `workgroup` parameters in the `/etc/samba/smb.conf` file. Otherwise, polling results may contain duplicate hosts.

IP range polling by using a distribution point

To configure IP range polling by using the distribution point:

1. Open the distribution point properties.
2. Go to the **IP range polling** section, and then select the **Enable range polling** option.

   The **IP range** window opens.
3. Specify the name of a new IP range.
4. Click **Add**, and then specify the IP range by using the address and subnet mask, or by using the start and end IP address. You can also add an existing subnet by clicking the **Browse** button.
5. Click the **Set polling schedule** button to specify the polling schedule options, if needed.

   Polling starts only according to the specified schedule. A manual start of polling is not available.

Polling schedule options:

- **Every N days**

  The polling runs regularly, with the specified interval in days, starting from the specified date and time.

  By default, the polling runs every day, starting from the current system date and time.

- **Every N minutes**

  The polling runs regularly, with the specified interval in minutes, starting from the specified time.

- **By days of week**

  The polling runs regularly, on the specified days of week, and at the specified time.

- **Every month on specified days of selected weeks**

  The polling runs regularly, on the specified days of each month, and at the specified time.

- **Run missed tasks**

  If the Administration Server is switched off or unavailable during the time for which the poll is scheduled, the Administration Server can either start the poll immediately after it is switched on, or wait for the next time for which the poll scheduled.

  If this option is enabled, the Administration Server starts polling immediately after it is switched on.

  If this option is disabled, the Administration Server waits for the next time for which the polling is scheduled.

  By default, this option is disabled.

6. Enable the **Use Zeroconf to poll IPv6 networks** option, to automatically poll the IPv6 network by using zero-configuration networking (also referred to as *Zeroconf*).

   In this case, the specified IP ranges are ignored because the distribution point polls the whole network. The **Use Zeroconf to poll IPv6 networks** option is available if the distribution point runs Linux. To use Zeroconf IPv6 polling, you must install the avahi-browse utility on the distribution point.

After the polling is completed, the newly discovered devices are automatically included in the **Managed devices** group, if you set up and enabled device moving rules. If no moving rules have been enabled, the newly discovered devices are automatically included in the **Unassigned devices** group.

Domain controller polling

Open Single Management Platform supports polling of the following domain controller types:

- Microsoft Active Directory
- Samba

- FreeIPA
- ALD Pro

For a Samba domain controller, Samba 4 is used as an Active Directory domain controller.

When you poll a domain controller, Administration Server or a distribution point retrieves information about the domain structure, user accounts, security groups, and DNS names of the devices that are included in the domain.

We recommend using domain controller polling if all networked devices are members of a domain. If some of the networked devices are not included in the domain, these devices cannot be discovered by domain controller polling.

Prerequisites

Before you poll a domain controller, ensure that the following protocols are enabled:

- Simple Authentication and Security Layer (SASL)
- Lightweight Directory Access Protocol (LDAP)

Ensure that the following ports are available on the domain controller device:

- 389 for SASL
- 636 for TLS

For all Linux-based hosts within a Microsoft Active Directory, Samba, FreeIPA, or ALD Pro domain controller, you have to specify the `netbios name` and `workgroup` parameters in the `/etc/samba/smb.conf` file. Otherwise, polling results may contain duplicate hosts.

> If a Microsoft Active Directory domain contains multiple domain controllers, selecting a specific controller during domain polling is not supported. The connection is established with the first detected domain controller. To ensure proper authentication with the Microsoft Active Directory domain, the Subject Alternative Name (SAN) field of the certificate must include all addresses of the domain controllers included in this domain, as well as the NetBIOS name of the Microsoft Active Directory domain.

Domain controller polling by using Administration Server

To poll a domain controller by using Administration Server:

1. In the main menu, go to **Discovery & deployment** → **Discovery** → **Domain controllers**.
2. Click **Polling settings**.

   The **Domain controller polling settings** window opens.
3. In the **Polling** field, switch the toggle button to the **Enabled** position.
4. In the **Poll specified domains**, click **Add**, then specify the address and user credentials of the domain controller, and click **OK**.

5. In the **Domain controller polling settings** window, do the following:

   - Select the type of the domain controller to be polled: **Active Directory or Samba 4**, **FreeIPA**, **ALD Pro**.

   - If necessary, set the polling schedule. The default period is one hour. The data received at the next polling completely replaces old data.

     The following polling schedule options are available:

     - **Every N days**

       The polling runs regularly, with the specified interval in days, starting from the specified date and time.

       By default, the polling runs every day, starting from the current system date and time.

     - **Every N minutes**

       The polling runs regularly, with the specified interval in minutes, starting from the specified time.

     - **By days of week**

       The polling runs regularly, on the specified days of week, and at the specified time.

     - **Every month on specified days of selected weeks**

       The polling runs regularly, on the specified days of each month, and at the specified time.

     - **Run missed tasks**

       If the Administration Server is switched off or unavailable during the time for which the poll is scheduled, the Administration Server can either start the poll immediately after it is switched on, or wait for the next time for which the poll scheduled.

       If this option is enabled, the Administration Server starts polling immediately after it is switched on.

       If this option is disabled, the Administration Server waits for the next time for which the polling is scheduled.

       By default, this option is disabled.

   If you change user accounts in a security group of the domain, these changes will be displayed in Open Single Management Platform an hour after you poll the domain controller.

6. If necessary, change the tenant to be assigned to the discovered user accounts. To do this, select the required tenant from the **Tenant** drop-down list.

   By default, the Administration Server tenant is specified. You can select several tenants.

7. Click **Save** to apply changes.

8. If you want to perform the poll immediately, click the **Start poll** button.

Domain controller polling by using a distribution point

You can also poll a domain controller by using a distribution point. A Windows- or Linux-based managed device can act as a distribution point.

> For a Linux distribution point, polling of all domain controller types is supported.
> For a Windows distribution point, only polling of a Microsoft Active Directory domain controller is supported.
> Polling with a Mac distribution point is not supported.

To configure domain controller polling by using the distribution point:

1. Open the distribution point properties.
2. Select the **Domain controller polling** section.
3. In the **Polling** field, switch the toggle button to the **Enabled** position.
4. If you use a Linux distribution point, select the type of the domain controller to be polled: **Active Directory or Samba 4**, **FreeIPA**, **ALD Pro**.

   This is not necessary that you specify the same domain controller type when configuring polling by using a distribution point and by using Administration Server.
5. Select the domain controller that you want to poll.

   If you use a Linux distribution point, in the **Poll specified domains** section, click **Add**, and then specify the address and user credentials of the domain controller.

   If you use a Windows distribution point, you can select one of the following options:

   - **Poll current domain**
   - **Poll entire domain forest**
   - **Poll specified domains**
6. Click the **Set polling schedule** button to specify the polling schedule options if needed.

   Polling starts only according to the specified schedule. Manual start of polling is not available.

After the polling is completed, the domain structure will be displayed in the **Domain controllers** section.

If you set up and enabled device moving rules, the newly discovered devices are automatically included in the **Managed devices** group. If no moving rules have been enabled, the newly discovered devices are automatically included in the **Unassigned devices** group.

The discovered user accounts can be used for domain authentication in OSMP Console over the Kerberos protocol.

Authentication and connection to a domain controller

On initial connection to the domain controller the Administration Server identifies the connection protocol. This protocol is used for all future connections to the domain controller.

The initial connection to a domain controller proceeds as follows:

1. Administration Server attempts to connect to the domain controller over TLS.

   By default, certificate verification is not required. Set the KLNAG_LDAP_TLS_REQCERT flag to 1 to enforce certificate verification.

   By default, the OS-dependent path to the certificate authority (CA) is used to access the certificate chain. Use the KLNAG_LDAP_SSL_CACERT flag to specify a custom path.

2. If the TLS connection fails, Administration Server attempts to connect to the domain controller over SASL (DIGEST-MD5).

3. If the SASL (DIGEST-MD5) connection fails, Administration Server uses Simple Authentication over non-encrypted TCP connection to connect to the domain controller.

You can use the KDT command to configure flags. For example, you can enforce certificate verification. To do this, on the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv klserver -n
KLNAG_LDAP_TLS_REQCERT -t d -v 1'
```

Configuring a Samba domain controller

Open Single Management Platform supports a Linux domain controller running only on Samba 4.

A Samba domain controller supports the same schema extensions as a Microsoft Active Directory domain controller. You can enable full compatibility of a Samba domain controller with a Microsoft Active Directory domain controller by using the Samba 4 schema extension. This is an optional action.

> We recommend enabling full compatibility of a Samba domain controller with a Microsoft Active Directory domain controller. This will ensure the correct interaction between Open Single Management Platform and the Samba domain controller.

To enable full compatibility of a Samba domain controller with a Microsoft Active Directory domain controller:

1. Execute the following command to use the RFC2307 schema extension:

   ```
   samba-tool domain provision --use-rfc2307 --interactive
   ```

2. Enable the schema update in a Samba domain controller. To do this, add the following line to the `/etc/samba/smb.conf` file:

   ```
   dsdb:schema update allowed = true
   ```

   > If the schema update completes with an error, you need to perform a full restore of the domain controller that acts as a schema master.

For all Linux-based hosts within a Samba domain controller, specify the `netbios name` and `workgroup` parameters in the `/etc/samba/smb.conf` file. Otherwise, polling results may contain duplicate hosts.

Using VDI dynamic mode on client devices

A virtual infrastructure can be deployed on a corporate network using temporary virtual machines. Open Single Management Platform detects temporary virtual machines and adds information about them to the Administration Server database. After a user finishes using a temporary virtual machine, the machine is removed from the virtual infrastructure. However, a record about the removed virtual machine can be saved in the database of the Administration Server. Also, nonexistent virtual machines can be displayed in OSMP Console.

To prevent information about nonexistent virtual machines from being saved, Open Single Management Platform supports dynamic mode for Virtual Desktop Infrastructure (VDI). The administrator can enable support of dynamic mode for VDI in the properties of the installation package of Network Agent to be installed on the temporary virtual machine.

When a temporary virtual machine is disabled, Network Agent notifies the Administration Server that the machine has been disabled. If the virtual machine has been disabled successfully, it is removed from the list of devices connected to the Administration Server. If the virtual machine is disabled with errors and Network Agent does not send a notification about the disabled virtual machine to the Administration Server, a backup scenario is used. In this scenario, the virtual machine is removed from the list of devices connected to the Administration Server after three unsuccessful attempts to synchronize with the Administration Server.

Enabling VDI dynamic mode in the properties of an installation package for Network Agent

To enable VDI dynamic mode:

1. In the main menu, go to **Discovery & deployment** → **Deployment & assignment** → **Installation packages**.
2. In the context menu of the Network Agent installation package, select **Properties**.
   The **Properties** window opens.
3. In the **Properties** window, select the **Advanced** section.
4. In the **Advanced** section, select the **Enable dynamic mode for VDI** option.

The device on which Network Agent is to be installed becomes a part of VDI.

Moving devices from VDI to an administration group

To move devices that are part of VDI to an administration group:

1. Go to **Asset management** → **Moving rules**.
2. Click **Add**.
3. On the **Rule conditions** tab, select the **Virtual machines** tab.
4. Set the **This is a virtual machine** rule to **Yes** and **Part of Virtual Desktop Infrastructure** to **Yes.**
5. Click **Save**.

Managing client devices

Kaspersky Next XDR Expert allows you to manage client devices:

- View settings and statuses of managed devices, including clusters and server arrays.
- Configure distribution points.
- Manage tasks.

You can use administration groups to combine client devices in a set that can be managed as a single unit. A client device can be included in only one administration group. Devices can be allocated to a group automatically based on **Rule conditions**:

- Creating device moving rules.
- Copying device moving rules.
- Conditions for a device moving rule.

You can use device selections to filter devices based on a condition. You can also tag devices for creating selections, for finding devices, and for distributing devices among administration groups.

Settings of a managed device

To view the settings of a managed device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The list of managed devices is displayed.

2. In the list of managed devices, click the link with the name of the required device.

The properties window of the selected device is displayed.

The following tabs are displayed in the upper part of the properties window representing the main groups of the settings:

- **General**

  This tab comprises the following sections:

  - The **General** section displays general information about the client device. Information is provided on the basis of data received during the last synchronization of the client device with the Administration Server:

    - **Name**

      In this field, you can view and modify the client device name in the administration group.

    - **Description**

      In this field, you can enter an additional description for the client device.

    - **Device status**

      Status of the client device assigned on the basis of the criteria defined by the administrator for the status of anti-virus protection on the device and the activity of the device on the network.

    - **Device owner**

      Name of the device owner. You can assign or remove a user as a device owner by clicking the **Manage device owner** link.

    - **Full group name**

      Administration group, which includes the client device.

    - **Last update of anti-virus databases**

      Date the anti-virus databases or applications were last updated on the device.

    - **Connected to Administration Server**

      Date and time Network Agent installed on the client device last connected to the Administration Server.

    - **Last visible**

      Date and time the device was last visible on the network.

- **Network Agent version**

  Version of the installed Network Agent.

- **Created**

  Date of the device creation within Open Single Management Platform.

- **Do not disconnect from the Administration Server**

  If this option is enabled, continuous connectivity between the managed device and the Administration Server is maintained. You may want to use this option if you are not using push servers, which provide such connectivity.

  If this option is disabled and push servers are not in use, the managed device only connects to the Administration Server to synchronize data or to transmit information.

  The maximum total number of devices with the **Do not disconnect from the Administration Server** option selected is 300.

  This option is disabled by default on managed devices. This option is enabled by default on the device where the Administration Server is installed and stays enabled even if you try to disable it.

- **Sessions**

  This section contains a table with information about the users currently logged in to the device. The table displays the following data in the corresponding columns:

  - User name
  - SAM account name
  - User principal name
  - Email address

  To correctly display information about domain users' sessions, you must use domain controller polling.

- The **Network** section displays the following information about the network properties of the client device:

  - **IP address**

    Device IP address.

  - **Windows domain**

    Workgroup that contains the device.

- **DNS name**

  Name of the DNS domain of the client device.

- **NetBIOS name**

  Name of the client device.

- **IPv6 address**

- The **System** section provides information about the operating system installed on the client device:

  - **OS**

  - **CPU architecture**

  - **Device name**

  - **Virtual machine type**

    The virtual machine manufacturer.

  - **Dynamic virtual machine as part of VDI**

    This row displays whether the client device is a dynamic virtual machine as part of VDI.

- The **Protection** section provides the following information about the current status of anti-virus protection on the client device:

  - **Visible**

    Visibility status of the client device.

  - **Device status**

    Status of the client device assigned on the basis of the criteria defined by the administrator for the status of anti-virus protection on the device and the activity of the device on the network.

  - **Status description**

    Status of the client device protection and connection to Administration Server.

  - **Protection status**

    This field shows the current status of real-time protection on the client device.

    When the status changes on the device, the new status is displayed in the device properties window only after the client device is synchronized with the Administration Server.

- **Last full scan**

  Date and time the last malware scan was performed on the client device.

- **Virus detected**

  Total number of threats detected on the client device since installation of the security application (first scan), or since the last reset of the threat counter.

- **Objects that have failed disinfection**

  Number of unprocessed files on the client device.

  This field ignores the number of unprocessed files on mobile devices.

- **Disk encryption status**

  The current status of file encryption on the local drives of the device. For a description of the statuses, see the Kaspersky Endpoint Security for Windows Help.

  Files can be only encrypted on the managed devices on which Kaspersky Endpoint Security for Windows is installed.

- The **Device status defined by application** section provides information about the device status that is defined by the managed application installed on the device. This device status can differ from the one defined by Open Single Management Platform.

- **Applications**

  This tab lists all Kaspersky applications installed on the client device.This tab contains the **Start** and **Stop** buttons that allow you to start and stop the selected Kaspersky application (excluding Network Agent). You can use these buttons if port 15000 UDP is available on the managed device for receipt push-notifications from Administration Server. If the managed device is unavailable for push-notifications, but the mode of continuous connection to Administration Server is enabled (the **Do not disconnect from the Administration Server** option in the **General** section is enabled), the **Start** and **Stop** buttons are available too. Otherwise, when you try to start or stop the application, an error message is displayed. Also you can click the application name to view general information about the application, a list of events that have occurred on the device, and the application settings.

- **Active policies and policy profiles**

  This tab lists the policies and policy profiles that are currently assigned to the managed device.

- **Tasks**

  On the **Tasks** tab, you can manage client device tasks: view the list of existing tasks, create new ones, remove, start and stop tasks, modify their settings, and view execution results. The list of tasks is provided based on data received during the last session of client synchronization with the Administration Server. The Administration Server requests the task status details from the client device. If port 15000 UDP is available on the managed device for receipt push-notifications from Administration Server, the task status is displayed and buttons for managing the task are enabled. If a push server cannot reach the managed device, but the mode of continuous connection to Administration Server is enabled (the **Do not disconnect from the Administration Server** option in the **General** section is enabled), the actions with tasks are available too.

  If connection is not established, the status is not displayed and buttons are disabled.

- **Events**

  The **Events** tab displays events logged on the Administration Server for the selected client device.

  You can export, copy, and delete the device events.

- **Security issues**

  In the **Security issues** tab, you can view, edit, and create security issues for the client device. Security issues can be created either automatically, through managed Kaspersky applications installed on the client device, or manually by the administrator. For example, if some users regularly move malware from their removable drives to devices, the administrator can create a security issue. The administrator can provide a brief description of the case and recommended actions (such as disciplinary actions to be taken against a user) in the text of the security issue, and can add a link to the user or users.

  A security issue for which all of the required actions have been taken is called *processed*. The presence of unprocessed security issues can be chosen as the condition for a change of the device status to *Critical* or *Warning*.

  This section contains a list of security issues that have been created for the device. Security issues are classified by severity level and type. The type of a security issue is defined by the Kaspersky application, which creates the security issue. You can highlight processed security issues in the list by selecting the check box in the **Processed** column.

- **Tags**

  In the **Tags** tab, you can manage the list of keywords that are used for finding client devices: view the list of existing tags, assign tags from the list, configure auto-tagging rules, add new tags and rename old tags, and remove tags.

- **Advanced**

  This tab comprises the following sections:

  - **Applications registry**. In this section, you can view the registry of applications installed on the client device and their updates; you can also set up the display of the applications registry.

    Information about installed applications is provided if Network Agent installed on the client device sends required information to the Administration Server. You can configure sending of information to the Administration Server in the properties window of Network Agent or its policy, in the **Repositories** section.

    Clicking an application name opens a window that contains the application details and a list of the update packages installed for the application.

  - **Executable files**. This section displays executable files found on the client device.

  - **Distribution points**. This section provides a list of distribution points with which the device interacts.

    - **Export to file**

      Click the **Export to file** button to save to a file a list of distribution points with which the device interacts. By default, the application exports the list of devices to a CSV file.

    - **Properties**

      Click the **Properties** button to view and configure the distribution point with which the device interacts.

  - **Hardware registry**. In this section, you can view information about hardware installed on the client device.

    If Network Agent is installed on a device running Windows, it sends to the Administration Server the following information about the device hardware:

    - RAM
    - Mass storage devices
    - Motherboard
    - CPU
    - Network adapters
    - Monitors
    - Video adapter
    - Sound card

If Network Agent is installed on a device running Linux or macOS, it sends to the Administration Server the following information about the device hardware, if this information is provided by the operating system:

- Total RAM volume
- Total volume of mass storage devices
- Motherboard
- CPU
- Network adapters

- **Available updates**. This section displays a list of software updates found on this device but not installed yet.

- **Software vulnerabilities**. This section provides information about vulnerabilities in third-party applications installed on client devices.

  To save the vulnerabilities to a file, select the check boxes next to the vulnerabilities that you want to save, and then click the **Export to CSV** button or **Export to TXT** button.

  The section contains the following settings:

  - **Show only vulnerabilities that can be fixed**

    If this option is enabled, the section displays vulnerabilities that can be fixed by using a patch.

    If this option is disabled, the section displays both vulnerabilities that can be fixed by using a patch, and vulnerabilities for which no patch has been released.

    By default, this option is enabled.

  - **Vulnerability properties**

    Click a software vulnerability name in the list to view the properties of the selected software vulnerability in a separate window. In the window, you can do the following:

    - Ignore software vulnerability on this managed device (in OSMP Console).
    - View the list of recommended fixes for the vulnerability.
    - Manually specify the software updates to fix the vulnerability (in OSMP Console).
    - View vulnerability instances.
    - View the list of existing tasks to fix vulnerability and create new tasks to fix vulnerability.

- **Remote diagnostics**. In this section, you can perform remote diagnostics of client devices.

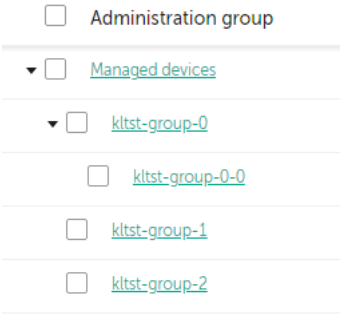
If you use a PostgreSQL, MariaDB or MySQL DBMS, the **Events** tab may display an incomplete list of events for the selected client device. This occurs when the DBMS stores a very large amount of events. You can increase the number of displayed events by doing either of the following:

- Removing unnecessary events.
- Reducing the storage term for unnecessary events.

To see a full list of events logged on the Administration Server for the device, use Reports.

Creating administration groups

Immediately after Open Single Management Platform installation, the hierarchy of administration groups contains only one administration group called **Managed devices**. When creating a hierarchy of administration groups, you can add devices and virtual machines to the **Managed devices** group, and add nested groups (see the figure below).



Viewing administration groups hierarchy

To create an administration group:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.
2. In the administration group structure, select the administration group that is to include the new administration group.
3. Click the **Add** button.
4. In the **Name of the new administration group** window that opens, enter a name for the group, and then click the **Add** button.

A new administration group with the specified name appears in the hierarchy of administration groups.

To create a structure of administration groups:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.
2. Click the **Import** button.

The New Administration Group Structure Wizard starts. Follow the instructions of the Wizard.

Device moving rules

We recommend that you automate the allocation of devices to administration groups through *device moving rules*. A device moving rule consists of three main parts: a name, an execution condition (logical expression with the device attributes), and a target administration group. A rule moves a device to the target administration group if the device attributes meet the rule execution condition.

All device moving rules have priorities. The Administration Server checks the device attributes as to whether they meet the execution condition of each rule, in ascending order of priority. If the device attributes meet the execution condition of a rule, the device is moved to the target group, so the rule processing is complete for this device. If the device attributes meet the conditions of multiple rules, the device is moved to the target group of the rule with the highest priority (that is, has the highest rank in the list of rules).

Device moving rules can be created implicitly. For example, in the properties of an installation package or a remote installation task, you can specify the administration group to which the device must be moved after Network Agent is installed on it. Also, device moving rules can be created explicitly by the administrator of Open Single Management Platform, in the **Asset management** → **Moving rules** section.

By default, a device moving rule is intended for one-time initial allocation of devices to administration groups. The rule moves devices from the unassigned devices group only once. If a device once was moved by this rule, the rule will never move it again, even if you return the device to the unassigned devices group manually. This is the recommended way of applying moving rules.

You can move devices that have already been allocated to some of the administration groups. To do this, in the properties of a rule, clear the **Move only devices that do not belong to an administration group** check box.

Applying moving rules to devices that have already been allocated to some of the administration groups, significantly increases the load on the Administration Server.

The **Move only devices that do not belong to an administration group** check box is locked in the properties of automatically created moving rules. Such rules are created when you add the *Install application remotely* task or create a stand-alone installation package.

You can create a moving rule that would affect a single device repeatedly.

We strongly recommend that you avoid moving a single device from one group to another repeatedly (for example, in order to apply a special policy to that device, run a special group task, or update the device through a specific distribution point).

Such scenarios are not supported, because they increase the load on Administration Server and network traffic to an extreme degree. These scenarios also conflict with the operating principles of Open Single Management Platform (particularly in the area of access rights, events, and reports). Another solution must be found, for example, through the use of policy profiles, tasks for device selections, assignment of Network Agents according to the standard scenario.

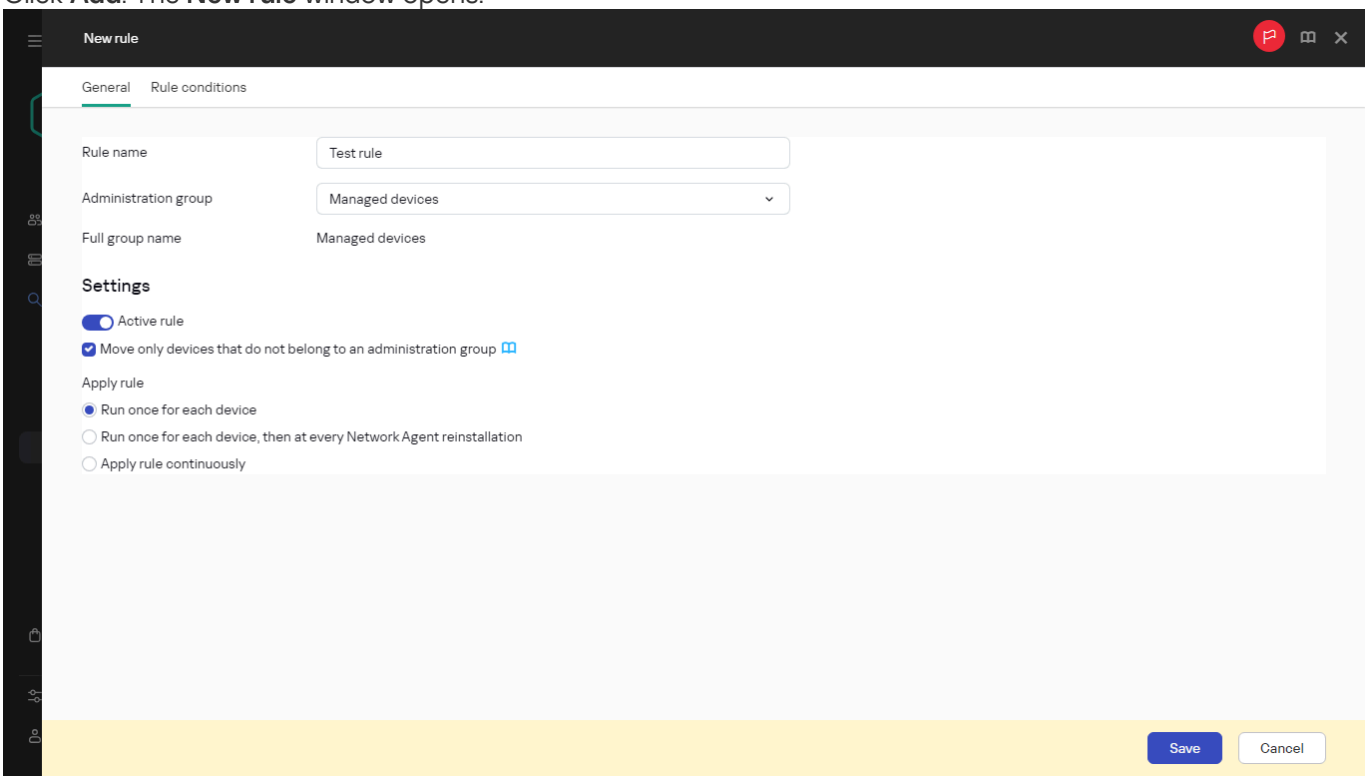
Creating device moving rules

You can set up device moving rules, that is, rules that automatically allocate devices to administration groups.

To create a moving rule:

1. In the main menu, go to **Asset management** → **Moving rules**.

2. Click **Add**. The **New rule** window opens.



Creating a new moving rule

3. In the window that opens, specify the following information on the **General** tab:

- **Rule name**

  Enter a name for the new rule.

  If you are copying a rule, the new rule gets the same name as the source rule, but an index in () format is added to the name, for example: (1).

- **Administration group**

  Select the administration group into which the devices are to be moved automatically.

- **Active rule**

  If this option is enabled, the rule is enabled and starts working after it is saved.

  If this option is disabled, the rule is created, but not enabled. It will not work until you enable this option.

- **Move only devices that do not belong to an administration group**

  If this option is enabled, only unassigned devices will be moved to the selected group.

If this option is disabled, devices that already belong to other administration groups, as well as unassigned devices, will be moved to the selected group.

- **Apply rule**

  You can select one of the following options:

  - **Run once for each device**

    The rule is applied once for each device that matches your criteria.

  - **Run once for each device, then at every Network Agent reinstallation**

    The rule is applied once for each device that matches your criteria, then only when Network Agent is reinstalled on these devices.

  - **Apply rule continuously**

    The rule is applied according to the schedule which the Administration Server sets up automatically (usually every several hours).

4. On the **Rule conditions** tab, specify at least one criterion by which the devices are moved to an administration group.

5. Click **Save**.

The moving rule is created. It is displayed in the list of moving rules.

The higher the position is on the list, the higher the priority of the rule. To increase or decrease the priority of a moving rule, move the rule up or down in the list, respectively, by using the mouse.

If the **Apply rule continuously** option is selected, the moving rule is applied regardless of the priority settings. Such rules are applied according to the schedule that the Administration Server sets up automatically.

If the device attributes meet the conditions of multiple rules, the device is moved to the target group of the rule with the highest priority (that is, has the highest rank in the list of rules).

Copying device moving rules

You can copy moving rules, for example, if you want to have several identical rules for different target administration groups.

To copy an existing a moving rule:

1. Do one of the following:

   - In the main menu, go to **Asset management** → **Moving rules**.

   - In the main menu, go to **Discovery & deployment** → **Deployment & assignment** → **Moving rules**.

   The list of moving rules is displayed.

2. Select the check box next to the rule you want to copy.

3. Click **Copy**.

4. In the window that opens, change the following information on the **General** tab—or make no changes if you only want to copy the rule without changing its settings:

   - **Rule name**

     Enter a name for the new rule.

     If you are copying a rule, the new rule gets the same name as the source rule, but an index in () format is added to the name, for example: (1).

   - **Administration group**

     Select the administration group into which the devices are to be moved automatically.

   - **Active rule**

     If this option is enabled, the rule is enabled and starts working after it is saved.

     If this option is disabled, the rule is created, but not enabled. It will not work until you enable this option.

   - **Move only devices that do not belong to an administration group**

     If this option is enabled, only unassigned devices will be moved to the selected group.

     If this option is disabled, devices that already belong to other administration groups, as well as unassigned devices, will be moved to the selected group.

   - **Apply rule**

     You can select one of the following options:

     - **Run once for each device**

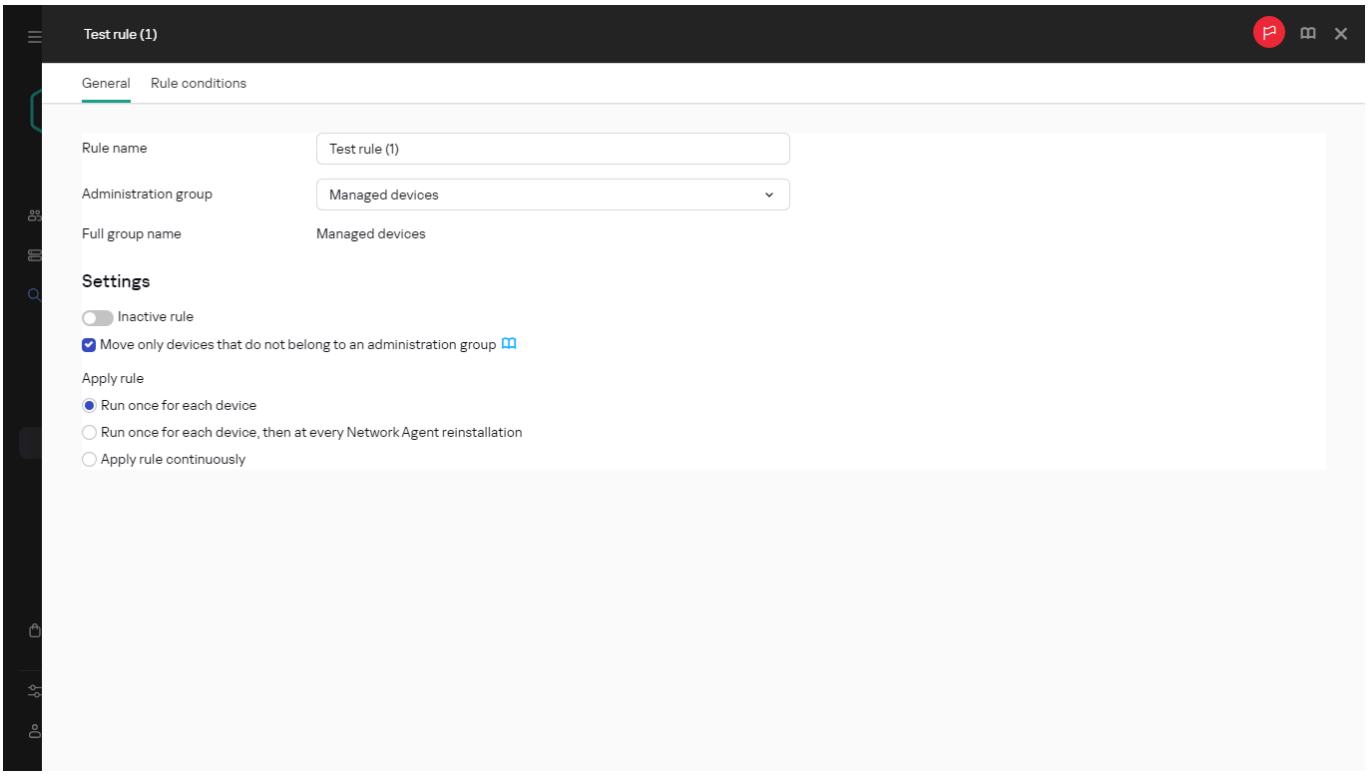
       The rule is applied once for each device that matches your criteria.

     - **Run once for each device, then at every Network Agent reinstallation**

       The rule is applied once for each device that matches your criteria, then only when Network Agent is reinstalled on these devices.

     - **Apply rule continuously**

       The rule is applied according to the schedule which the Administration Server sets up automatically (usually every several hours).



Properties of the device moving rule

5. On the **Rule conditions** tab, specify at least one criterion for the devices that you want to be moved automatically.

6. Click **Save**.

The new moving rule is created. It is displayed in the list of moving rules.

Conditions for a device moving rule

When you create or copy a rule to move client devices to administration groups, on the **Rule conditions** tab you set conditions for moving the devices. To determine which devices to move, you can use the following criteria:

- Tags assigned to client devices.
- Network parameters. For example, you can move devices with IP addresses from a specified range.
- Managed applications installed on client devices, for instance, Network Agent or Administration Server.
- Virtual machines, which are the client devices.

Below, you can find the description on how to specify this information in a device moving rule.

If you specify several conditions in the rule, the AND logical operator works and all the conditions apply at the same time. If you do not select any options or keep some fields blank, such conditions do not apply.

Tags tab

On this tab, you can configure a device moving rule based on device tags that were previously added to the descriptions of client devices. To do this, select the required tags. Also, you can enable the following options:

- **Apply to devices without the specified tags**

  If this option is enabled, all devices with the specified tags are excluded from a device moving rule. If this option is disabled, the device moving rule applies to devices with all the selected tags.

  By default, this option is disabled.

- **Apply if at least one specified tag matches**

  If this option is enabled, a device moving rule applies to client devices with at least one of the selected tags. If this option is disabled, the device moving rule applies to devices with all the selected tags.

  By default, this option is disabled.

Network tab

On this tab, you can specify the network data of devices that a device moving rule considers:

- **DNS name of the device**

  DNS domain name of the client device that you want to move. Fill this field if your network includes a DNS server.

  If case sensitive collation is set for the database that you use for Open Single Management Platform, keep case when you specify a device DNS name. Otherwise, the device moving rule will not work.

- **DNS domain**

  A device moving rule applies to all devices included in the specified main DNS suffix. Fill this field if your network includes a DNS server.

- **IP range**

  If this option is enabled, you can enter the initial and final IP addresses of the IP range in which the relevant devices must be included.

  By default, this option is disabled.

- **IP address for connection to Administration Server**

  If this option is enabled, you can set the IP addresses by which client devices are connected to Administration Server. To do this, specify the IP range that includes all necessary IP addresses.

  By default, this option is disabled.

- **Device is in IP range**

  If this option is enabled, you can select an IP range that you previously added in the **IP ranges** section. The relevant devices must be included in the selected IP range.

  By default, this option is disabled.

- **Connection profile changed**

  Select one of the following values:

  - **Yes**. A device moving rule only applies to client devices with a changed connection profile.
  - **No**. The device moving rule only applies to the client devices whose connection profile has not changed.
  - **No value is selected**. The condition does not apply.

- **Managed by a different Administration Server**

  Select one of the following values:

  - **Yes**. A device moving rule only applies to client devices managed by other Administration Servers. These Servers are different from the Server on which you configure the device moving rule.
  - **No**. The device moving rule only applies to client devices managed by the current Administration Server.
  - **No value is selected**. The condition does not apply.

Device owner tab

On this tab, you can configure a device moving rule based on the device owner, security group membership, and role:

- **Device owner**

  Select a device owner's user name from an internal security group. Learn more about users and user roles in this section.

  No more than one user can be registered as the device owner.

- **Device owner's membership in Active Directory security group**

  Select an external Active Directory security group that the device owner belongs to.

  The user can be a part of an Active Directory security group or a part of a group that is included in this Active Directory security group.

- **Device owner's role**

  Select the device owner's assigned role. Learn more about user roles in this article.

- **Device owner's membership in an internal security group**

  Select an internal security group that the device owner belongs to.

Applications tab

On this tab, you can configure a device moving rule based on the managed applications and operating systems installed on client devices:

- **Network Agent is installed**

  Select one of the following values:

  - **Yes**. A device moving rule only applies to client devices with Network Agent installed.
  - **No**. The device moving rule only applies to client devices on which Network Agent is not installed.
  - **No value is selected**. The condition does not apply.

- **Applications**

  Specify what managed applications should be installed on client devices, so a device moving rule applies to these devices. For example, you can select **Kaspersky Security Center 16.1 Network Agent** or **Kaspersky Security Center 16.1 Administration Server**.

  If you do not select any managed application, the condition does not apply.

- **Enabled**

  You can cull client devices based on the operating system version. For this purpose, specify operating systems that should be installed on the client devices. As a result, a device moving rule applies to the client devices with the selected operating systems.

  If you do not enable this option, the condition does not apply. By default, the option is disabled.

- **Operating system architecture**

  You can cull client devices by the operating system bit sizes. In the **Operating system architecture** field, you can select one of the following values:

  - **Unknown**
  - **x86**
  - **AMD64**
  - **IA64**

To check the operating system bit size of the client devices:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section.
2. Click the settings icon (⚙) on the right.
3. Select the **OS architecture** option, and then click the **Save** button.

   After that, the operating system bit size is displayed for every managed device.

- **OS package manager**

You can cull client devices by the operating system package manager. In the **OS package manager** field, you can select one of the following values:

- rpm—for devices running Red Hat Enterprise Linux, CentOS, Fedora, SUSE, Oracle Linux operating systems
- apt (dpkg)—for devices running Debian, Ubuntu, Kali operating systems
- pkg—for devices running FreeBSD, OpenBSD operating systems

To check the operating system package manager of the client devices:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section.
2. Click the **Columns settings** button (⚙) on the right.
3. Select the **OS package manager** option, and then click the **Save** button.

The operating system package manager is displayed for every managed device.

- **OS architecture**

In this field, you can specify the package version of the operating system (in the *X.Y* format), which will determine how the moving rule is applied to the device. By default, no version value is specified.

- **User certificate**

Select one of the following values:

- **Installed**. A device moving rule only applies to mobile devices with a mobile certificate.
- **Not installed**. The device moving rule only applies to mobile devices without a mobile certificate.
- **No value is selected**. The condition does not apply.

- **Enabled**

This setting is applicable to Windows operating systems only.

You can specify whether the selected operating system must have an equal, earlier, or later build number. You can also configure a device moving rule for all build numbers except the specified one.

- **Enabled**

  This setting is applicable to Windows operating systems only.

  You can specify whether the selected operating system must have an equal, earlier, or later release number. You can also configure a device moving rule for all release numbers except the specified one.

Virtual machines tab

On this tab, you can configure a device moving rule according to whether client devices are virtual machines or part of a virtual desktop infrastructure (VDI):

- **This is a virtual machine**

  In the drop-down list, you can select one of the following:

  - **N/A**. The condition does not apply.
  - **No**. Move devices that are not virtual machines.
  - **Yes**. Move devices that are virtual machines.

- **Virtual machine type**
- **Part of Virtual Desktop Infrastructure**

  In the drop-down list, you can select one of the following:

  - **N/A**. The condition does not apply.
  - **No**. Move devices that are not part of VDI.
  - **Yes**. Move devices that are part of VDI.

Domain controller tab

On this tab, you can specify that it is necessary to move devices included in the domain organizational unit. You can also move devices from all child organizational units of the specified domain organizational unit:

- **Device is included in the following organizational unit**

  If this option is enabled, a device moving rule applies to devices from the domain controller organizational unit specified in the list under the option.

  By default, this option is disabled.

- **Include child organizational units**

  If this option is enabled, the selection includes devices from all child organizational units of the specified domain controller organizational unit.

  By default, this option is disabled.

- **Move devices from child units to corresponding subgroups**
- **Create subgroups corresponding to containers of newly detected devices**
- **Delete subgroups that are not present in the domain**
- **Device is included in the following domain security group**

  If this option is enabled, a device moving rule applies to devices from the domain security group specified in the list under the option.

  By default, this option is disabled.

Adding devices to an administration group manually

You can move devices to administration groups automatically by creating device moving rules or manually by moving devices from one administration group to another or by adding devices to a selected administration group. This section describes how to manually add devices to an administration group.

To add manually one or more devices to a selected administration group:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.
2. Select the administration group to which you want to add the devices:
   - For the root group:

     In this case you can proceed to the next step.
   - For a subgroup:

     a. Click the **Show hierarchy** button at the top of the page.

     The button name changes to **Hide hierarchy**.

     b. In the window that opens, click the name of the subgroup.

     The path to the selected group is displayed at the top of the page. If necessary, you can click a link with the administration group name to go to the group. By default, the last link in the path is inactive. Also, you can close the hierarchy window by clicking the **Hide hierarchy** button.
3. Click the **Add devices** button.

   The Move devices wizard starts.
4. Make a list of the devices that you want to add to the administration group.

> You can add only devices for which information has already been added to the Administration Server database either upon connection of the device or after device discovery.

Select how you want to add devices to the list:

- Click the **Add devices** button, and then specify the devices in one of the following ways:
  - Select devices from the list of devices detected by the Administration Server.
  - Specify a device IP address or an IP range.

    > The IP range must not contain more than 1,048,576 addresses.

  - Specify a device DNS name.

    > The device name field must not contain space characters, backspace characters, or the following prohibited characters: , \ / * ' " ; : & ` ~ ! @ # $ ^ ( ) = + [ ] { } | < > %

- Click the **Import devices from file** button to import a list of devices from a .txt file. Each device address or name must be specified on a separate line.

  > The file must not contain space characters, backspace characters, or the following prohibited characters: , \ / * ' " ; : & ` ~ ! @ # $ ^ ( ) = + [ ] { } | < > %

5. View the list of devices to be added to the administration group. You can edit the list by adding or removing devices.

6. After making sure that the list is correct, click the **Next** button.

   The wizard processes the device list and displays the result. The successfully processed devices are added to the administration group and are displayed in the list of devices under names generated by Administration Server.

Moving devices or clusters to an administration group manually

You can move devices from one administration group to another, or from the group of unassigned devices to an administration group.

You can also move clusters or server arrays from one administration group to another. When you move a cluster or server array to another group, all of its nodes move with it, because a cluster and any of its nodes always belong to the same administration group. When you select a single cluster node on the **Devices** tab, the **Move to group** button becomes unavailable.

To move one or several devices or clusters to a selected administration group:

1. Open the administration group from which you want to move the devices. To do this, perform one of the following:

   - To open an administration group, in the main menu, go to **Asset management** → **Assets (Devices)**, and then select the required administration group.

     - For the root group, you can proceed to the next step.

     - For a subgroup, click the **Show hierarchy** button at the top of the page, and then, in the window that opens, select a required administration group.

   - To open the **Unassigned devices** group, in the main menu, go to **Discovery & deployment** → **Unassigned devices**.

2. If the administration group contains clusters or server arrays, the **Assets (Devices)** section is divided into two tabs—the **Devices** tab and the **Clusters and server arrays** tab. Open the tab for the object that you want to move.

3. Select the check boxes next to the devices or clusters that you want to move to a different group.

4. Click the **Move to group** button.

5. In the hierarchy of administration groups, select the check box next to the administration group to which you want to move the selected devices or clusters.

6. Click the **Move** button.

The selected devices or clusters are moved to the selected administration group.

About clusters and server arrays

Open Single Management Platform supports cluster technology. If Network Agent sends information to Administration Server confirming that an application installed on a client device is part of a server array, this client device becomes a cluster node.

If an administration group contains clusters or server arrays, the **Assets (Devices)** page displays two tabs—one for individual devices, and one for clusters and server arrays. After the managed devices are detected as cluster nodes, the cluster is added as an individual object to the **Clusters and server arrays** tab.

The cluster or server array nodes are listed on the **Devices** tab, along with other managed devices. You can view properties of the nodes as individual devices and perform other operations, but you cannot delete a cluster node or move it to another administration group separately from its cluster. You can only delete or move an entire cluster.

You can perform the following operations with clusters or server arrays:

- View properties

- Move the cluster or server array to another administration group

  When you move a cluster or server array to another group, all of its nodes move with it, because a cluster and any of its nodes always belong to the same administration group.

- Delete

  It is reasonable to delete a cluster or server array only when the cluster or server array does not exist in the organization network any longer. If a cluster is still visible on your network and Network Agent and the Kaspersky security application are still installed on the cluster nodes, Open Single Management Platform returns the deleted cluster and its nodes back to the list of managed devices automatically.

Properties of a cluster or server array

To view the settings of a cluster or server array:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Clusters and server arrays**.

   The list of clusters and server arrays is displayed.

2. Click the name of the required cluster or server array.

The properties window of the selected cluster or server array is displayed.

General

The **General** section displays general information about the cluster or server array. Information is provided on the basis of data received during the last synchronization of the cluster nodes with the Administration Server:

- **Name**
- **Description**
- **Windows domain**

  Windows domain or workgroup, which contains the cluster or server array.

- **NetBIOS name**

  Windows network name of the cluster or server array.

- **DNS name**

  Name of the DNS domain of the cluster or server array.

Tasks

In the **Tasks** tab, you can manage the tasks assigned to the cluster or server array: view the list of existing tasks; create new ones; remove, start, and stop tasks; modify task settings; and view execution results. The listed tasks relate to the Kaspersky security application installed on the cluster nodes. Open Single Management Platform receives the task list and the task status details from the cluster nodes. If a connection is not established, the status is not displayed.

Nodes

This tab displays a list of nodes included into the cluster or server array. You can click a node name to view the device properties window.

Kaspersky application

The properties window may also contain additional tabs with the information and settings related to the Kaspersky security application installed on the cluster nodes.

Adjustment of distribution points and connection gateways

A structure of administration groups in Open Single Management Platform performs the following functions:

- Sets the scope of policies

  There is an alternate way of applying relevant settings on devices, by using *policy profiles*.

- Sets the scope of group tasks

  There is an approach to defining the scope of group tasks that is not based on a hierarchy of administration groups: use of tasks for device selections and tasks for specific devices.

- Sets access rights to devices, virtual Administration Servers, and secondary Administration Servers

- Assigns distribution points

When building the structure of administration groups, you must take into account the topology of the organization's network for the optimum assignment of distribution points. The optimum distribution of distribution points allows you to save traffic on the organization's network.

Depending on the organizational schema and network topology, the following standard configurations can be applied to the structure of administration groups:

- Single office

- Multiple small remote offices

Devices functioning as distribution points must be protected, including physical protection, against any unauthorized access.

Standard configuration of distribution points: Single office

In a standard "single-office" configuration, all devices are on the organization's network so they can "see" each other. The organization's network may consist of a few separate parts (networks or network segments) linked by narrow channels.

The following methods of building the structure of administration groups are possible:

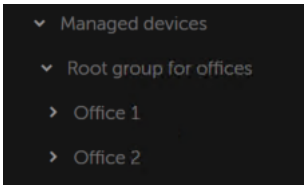
- Building the structure of administration groups taking into account the network topology. The structure of administration groups may not reflect the network topology with absolute precision. A match between the separate parts of the network and certain administration groups would be enough. You can use automatic assignment of distribution points or assign them manually.

- Building the structure of administration groups, without taking the network topology into account. In this case, you must disable automatic assignment of distribution points, and then assign one or several devices to act as distribution points for a root administration group in each of the separate parts of the network, for example, for the **Managed devices** group. All distribution points will be at the same level and will feature the same scope spanning all devices on the organization's network. In this case, each Network Agent will connect to the distribution point that has the shortest route. The route to a distribution point can be traced with the tracert utility.

Standard configuration of distribution points: Multiple small remote offices

This standard configuration provides for a number of small remote offices, which may communicate with the head office over the internet. Each remote office is located behind the NAT, that is, connection from one remote office to another is not possible because offices are isolated from one another.

The configuration must be reflected in the structure of administration groups: a separate administration group must be created for each remote office (groups **Office 1** and **Office 2** in the figure below).



Remote offices are included in the administration group structure

One or multiple distribution points must be assigned to each administration group that correspond to an office. Distribution points must be devices at the remote office that have a sufficient amount of free disk space. Devices deployed in the **Office 1** group, for example, will access distribution points assigned to the **Office 1** administration group.

If some users move between offices physically, with their laptops, you must select two or more devices (in addition to the existing distribution points) in each remote office and assign them to act as distribution points for a top-level administration group (**Root group for offices** in the figure above).

Example: A laptop is deployed in the **Office 1** administration group and then is moved physically to the office that corresponds to the **Office 2** administration group. After the laptop is moved, Network Agent attempts to access the distribution points assigned to the **Office 1** group, but those distribution points are unavailable. Then, Network Agent starts attempting to access the distribution points that have been assigned to the **Root group for offices**. Because remote offices are isolated from one another, attempts to access distribution points assigned to the **Root group for offices** administration group will only be successful when Network Agent attempts to access distribution points in the **Office 2** group. That is, the laptop will remain in the administration group that corresponds to the initial office, but the laptop will use the distribution point of the office where it is physically located at the moment.

Calculating the number and configuration of distribution points

The more client devices a network contains, the more distribution points it requires. We recommend that you not disable automatic assignment of distribution points. When automatic assignment of distribution points is enabled, Administration Server assigns distribution points if the number of client devices is quite large and defines their configuration.

Using exclusively assigned distribution points

If you plan to use certain specific devices as distribution points (that is, exclusively assigned servers), you can opt out of using automatic assignment of distribution points. In this case, make sure that the devices that you intend to make distribution points have sufficient volume of free disk space, are not shut down regularly, and have Sleep mode disabled.

Number of exclusively assigned distribution points on a network that contains a single network segment, based on the number of networked devices

Number of client devices in the network segment	Number of distribution points
Less than 300	0 (Do not assign distribution points)
More than 300	Acceptable: (N/10,000 + 1), recommended: (N/5000 + 2), where N is the number of networked devices

Number of exclusively assigned distribution points on a network that contains multiple network segments, based on the number of networked devices

Number of client devices per network segment	Number of distribution points
Less than 10	0 (Do not assign distribution points)
10–100	1
More than 100	Acceptable: (N/10,000 + 1), recommended: (N/5000 + 2), where N is the number of networked devices

Using standard client devices (workstations) as distribution points

If you plan to use standard client devices (that is, workstations) as distribution points, we recommend that you assign distribution points as shown in the tables below in order to avoid excessive load on the communication channels and on Administration Server:

Number of workstations functioning as distribution points on a network that contains a single network segment, based on the number of networked devices

Number of client devices in the network segment	Number of distribution points
Less than 300	0 (Do not assign distribution points)
More than 300	(N/300 + 1), where N is the number of networked devices; there must be at least 3 distribution points

Number of workstations functioning as distribution points on a network that contains multiple network segments, based on the number of networked devices

Number of client devices per network segment	Number of distribution points
Less than 10	0 (Do not assign distribution points)
10–30	1

Number of client devices per network segment	Number of distribution points
31–300	2
More than 300	(N/300 + 1), where N is the number of networked devices; there must be at least 3 distribution points

If a distribution point is shut down (or not available for some other reason), the managed devices in its scope can access the Administration Server for updates.

Assigning distribution points automatically

We recommend that you assign distribution points automatically. In this case, Open Single Management Platform will select on its own which devices must be assigned distribution points.

Devices running Microsoft Windows on Arm cannot act as distribution points.

To assign distribution points automatically:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Distribution points** section.

3. Select the **Automatically assign distribution points** option.

   If automatic assignment of devices as distribution points is enabled, you cannot configure distribution points manually or edit the list of distribution points.

4. Click the **Save** button.

   Administration Server assigns and configures distribution points automatically.

Assigning distribution points manually

Open Single Management Platform allows you to manually assign devices to act as distribution points.

We recommend that you assign distribution points automatically. In this case, Open Single Management Platform will select on its own which devices must be assigned distribution points. However, if you have to opt out of assigning distribution points automatically for any reason (for example, if you want to use exclusively assigned servers), you can assign distribution points manually after you Calculating the number and configuration of distribution points.

Devices functioning as distribution points must be protected, including physical protection, against any unauthorized access.

To manually assign a device to act as distribution point:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Distribution points** section.

3. Select the **Manually assign distribution points** option.

4. Click the **Assign** button.

5. Select the device that you want to make a distribution point.

   When selecting a device, keep in mind the operation features of distribution points and the requirements set for the device that acts as distribution point.

6. Select the administration group that you want to include in the scope of the selected distribution point.

7. Click the **OK** button.

   The distribution point that you have added will be displayed in the list of distribution points, in the **Distribution points** section.

8. Click the newly added distribution point in the list to open its properties window.

9. Configure the distribution point in the properties window:

   - The **General** section contains the settings of interaction between the distribution point and client devices.

     - **SSL port**

       The number of the SSL port for encrypted connection between client devices and the distribution point using SSL.

       By default, port 13000 is used.

     - **Use multicast**

       If this option is enabled, IP multicasting will be used for automatic distribution of installation packages to client devices within the group.

       IP multicasting decreases the time required to install an application from an installation package to a group of client devices, but increases the installation time when you install an application to a single client device.

     - **IP multicast address**

       IP address that will be used for multicasting. You can define an IP address in the range of 224.0.0.0 – 239.255.255.255

       By default, Open Single Management Platform automatically assigns a unique IP multicast address within the given range.

- **IP multicast port number**

  Number of the port for IP multicasting.

  By default, the port number is 15001. If the device with Administration Server installed is specified as the distribution point, port 13001 is used for SSL connection by default.

- **Distribution point address for remote devices**

  The network address through which remote devices connect to the distribution point.

- **Deploy updates**

  Updates are distributed to managed devices from the following sources:

  - This distribution point, if this option is enabled.
  - Other distribution points, Administration Server, or Kaspersky update servers, if this option is disabled.

  If you use distribution points to deploy updates, you can save traffic because you reduce the number of downloads. Also, you can relieve the load on the Administration Server and relocate the load between the distribution points. You can calculate the number of distribution points for your network to optimize the traffic and load.

  If you disable this option, the number of update downloads and load on the Administration Server may increase. By default, this option is enabled.

- **Deploy installation packages**

  Installation packages are distributed to managed devices from the following sources:

  - This distribution point, if this option is enabled.
  - Other distribution points, Administration Server, or Kaspersky update servers, if this option is disabled.

  If you use distribution points to deploy installation packages, you can save traffic because you reduce the number of downloads. Also, you can relieve the load on the Administration Server and relocate the load between the distribution points. You can calculate the number of distribution points for your network to optimize the traffic and load.

  If you disable this option, the number of installation package downloads and load on the Administration Server may increase. By default, this option is enabled.

- **Run push server**

  In Open Single Management Platform, a distribution point can work as a push server for the devices managed through the mobile protocol and for the devices managed by Network Agent. For example, a push server must be enabled if you want to be able to force synchronization of KasperskyOS devices with Administration Server. A push server has the same scope of managed devices as the distribution point on which the push server is enabled. If you have several distribution points assigned for the same administration group, you can enable push server on each of the distribution points. In this case, Administration Server balances the load between the distribution points.

- **Push server port**

  The port number for the push server. You can specify the number of any unoccupied port.

- In the **Scope** section, specify administration groups to which the distribution point will distribute updates.
- In the **Source of updates** section, you can select a source of updates for the distribution point:

  - **Source of updates**

    Select a source of updates for the distribution point:

    - To allow the distribution point to receive updates from the Administration Server, select **Retrieve from Administration Server**.
    - To allow the distribution point to receive updates by using a task, select **Use update download task**, and then specify a *Download updates to the repositories of distribution points* task:
      - If such a task already exists on the device, select the task in the list.
      - If no such task yet exists on the device, click the **Create task** link to create a task. The New task wizard starts. Follow the instructions of the wizard.

  - **Download diff files**

    This option enables the downloading diff files feature.

    By default, this option is enabled.

- If your distribution points use proxy server when connecting to the internet, in the **Internet connection settings** subsection, you can specify the following settings:

  - **Use proxy server**

    If this check box is selected, in the entry fields you can configure the proxy server connection.

    By default, this check box is cleared.

  - **Proxy server address**

    Address of the proxy server.

  - **Port number**

    Port number that is used for connection.

  - **Bypass proxy server for local addresses**

    If this option is enabled, no proxy server is used to connect to devices on the local network.

By default, this option is disabled.

- **Proxy server authentication**

  If this check box is selected, in the entry fields you can specify the credentials for proxy server authentication.

  By default, this check box is cleared.

- **User name**

  User account under which connection to the proxy server is established.

- **Password**

  Password of the account under which the task will be run.

- In the **KSN Proxy** section, you can configure the application to use the distribution point to forward KSN requests from the managed devices:

  - **Enable KSN Proxy on the distribution point side**

    The KSN proxy service is run on the device that is used as a distribution point. Use this feature to redistribute and optimize traffic on the network.

    The distribution point sends the KSN statistics, which are listed in the Kaspersky Security Network statement, to Kaspersky.

    By default, this option is disabled. Enabling this option takes effect only if the **Use Administration Server as a proxy server** and **I agree to use Kaspersky Security Network** options are enabled in the Administration Server properties window.

    You can assign a node of an active-passive cluster to a distribution point and enable KSN proxy server on this node.

  - **Forward KSN requests to Administration Server**

    The distribution point forwards KSN requests from the managed devices to the Administration Server.

    By default, this option is enabled.

  - **Access KSN Cloud/KPSN directly over the internet**

    The distribution point forwards KSN requests from managed devices to the KSN Cloud or KPSN. The KSN requests generated on the distribution point itself are also sent directly to the KSN Cloud or KPSN.

  - **Ignore proxy server settings when connecting to KPSN**

    Enable this option, if you have the proxy server settings configured in the distribution point properties or in the Network Agent policy, but your network architecture requires that you use KPSN directly. Otherwise, requests from the managed applications cannot reach KPSN.

This option is available if you select the **Access KSN Cloud/KPSN directly over the internet** option.

- **Port**

  The number of the TCP port that the managed devices will use to connect to KSN proxy server. The default port number is 13111.

- **Use UDP port**

  If you need the managed devices to connect to KSN proxy server through a UDP port, enable the **Use UDP port** option and specify a UDP port number. By default, this option is enabled.

- **UDP port**

  The number of the UDP port that the managed devices will use to connect to KSN proxy server. The default UDP port to connect to the KSN proxy server is 15111.

- **Use HTTPS**

  If you need the managed devices to connect to KSN proxy server through an HTTPS port, enable the **Use HTTPS** option and specify an **HTTPS through port** number. The default HTTPS port to connect to the KSN proxy server is 17111.

- **HTTPS port**

  The number of the HTTPS port that the managed devices will use to connect to KSN proxy server. The default HTTPS port to connect to the KSN proxy server is 17111.

- In the **Connection gateway** section, you can configure the distribution point to act as a gateway for connection between Network Agent instances and Administration Server:

  - **Connection gateway**

    If a direct connection between Administration Server and Network Agents cannot be established due to organization of your network, you can use the distribution point to act as the connection gateway between Administration Server and Network Agents.

    Enable this option if you need the distribution point to act as a connection gateway between Network Agents and Administration Server. By default, this option is disabled.

  - **Establish connection to gateway from Administration Server (if gateway is in DMZ)**

    If Administration Server is located outside the demilitarized zone (DMZ), on local area network, Network Agents installed on remote devices cannot connect to Administration Server. You can use a distribution point as the connection gateway with reverse connectivity (Administration Server establishes a connection to distribution point).

    Enable this option if you need to connect Administration Server to the connection gateway in DMZ.

- **Open local port for Kaspersky Security Center Web Console**

  Enable this option if you need the connection gateway in DMZ to open a port for Web Console that is in DMZ or on the internet. Specify the port number that will be used for the connection from Web Console to the distribution point. The default port number is 13299.

  This option is available if you enable the **Establish connection to gateway from Administration Server (if gateway is in DMZ)** option.

When connecting mobile devices to Administration Server via the distribution point that acts as a connection gateway, you can enable the following options:

- **Open port for mobile devices (SSL authentication of the Administration Server only)**

  Enable this option if you need the connection gateway to open a port for mobile devices and specify the port number that mobile devices will use for connection to distribution point. The default port number is 13292. The mobile device will check the Administration Server certificate. When establishing the connection, only Administration Server is authenticated.

- **Open port for mobile devices (two-way SSL authentication)**

  Enable this option if you need connection gateway to open a port that will be used for two-way authentication of Administration Server and mobile devices. Mobile device will check the Administration Server certificate, and Administration Server will check the mobile device certificate. Specify the following parameters:

  - Port number that mobile devices will use for connection to the distribution point. The default port number is 13293.

  - DNS domain names of the connection gateway that will be used by mobile devices. Separate domain names with commas. The specified domain names will be included in the distribution point certificate. If the domain names used by mobile devices do not match the common name in the distribution point certificate, mobile devices do not connect to the distribution point.

    The default DNS domain name is the FQDN name of the connection gateway.

In both cases, the certificates are checked during the TLS session establishment on distribution point only. The certificates are not forwarded to be checked by the Administration Server. After a TLS session with the mobile device is established, the distribution point uses the Administration Server certificate to create a tunnel for synchronization between the mobile device and Administration Server. If you open the port for two-way SSL authentication, the only way to distribute the mobile device certificate is via an installation package.

- Configure domain controller polling by the distribution point.

  - **Domain controller polling**

    You can enable device discovery for domain controllers.

    In the **Polling** field, after you switch the toggle button to the **Enabled** position, you can select domain controllers for polling and also specify the polling schedule for them.

    If you use a Linux distribution point, in the **Poll specified domains** section, click **Add**, and then specify the address and user credentials of the domain controller. Also, you can specify the type of domain to be polled: Active Directory or Samba, FreeIPA, ALD Pro.

If you use a Windows distribution point, you can select one of the following options:

- **Poll current domain**
- **Poll entire domain forest**
- **Poll specified domains**

- Configure the polling of IP ranges by the distribution point.
  - **IP ranges polling**

    You can enable device discovery for IPv4 ranges and IPv6 networks.

    If you enable the **Enable range polling** option, you can add scanned ranges and set the schedule for them. You can add IP ranges to the list of scanned ranges.

    If you enable the **Use Zeroconf to poll IPv6 networks** option, the distribution point automatically polls the IPv6 network by using zero-configuration networking (also referred to as *Zeroconf*). In this case, the specified IP ranges are ignored because the distribution point polls the whole network. The **Use Zeroconf to poll IPv6 networks** option is available if the distribution point runs Linux. To use Zeroconf IPv6 polling, you must install the avahi-browse utility on the distribution point.

- In the **Advanced** section, specify the folder that the distribution point must use to store distributed data.
  - **Use default folder**

    If you select this option, the application uses the Network Agent installation folder on the distribution point.

  - **Use specified folder**

    If you select this option, in the field below, you can specify the path to the folder. It can be a local folder on the distribution point, or it can be a folder on any device on the corporate network.

    The user account used on the distribution point to run Network Agent must have read/write access to the specified folder.

10. Click the **OK** button.

    The selected devices act as distribution points.

Modifying the list of distribution points for an administration group

You can view the list of distribution points assigned to a specific administration group and modify the list by adding or removing distribution points.

To view and modify the list of distribution points assigned to an administration group:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.
2. Select an administration group for which you want to view the assigned distribution points by clicking the **Show hierarchy** button at the top of the page, and then in the window that opens, click the name of the required group.

   This enables the **Distribution points** menu item.

   The path to the selected group is displayed at the top of the page. If necessary, in this path you can click a link with the administration group name to go to the group. By default, the last link in the path is inactive.
3. In the main menu, go to **Asset management** → **Distribution points**.
4. To add new distribution points for the administration group, click the **Assign** button.
5. To remove the assigned distribution points, select devices from the list and click the **Unassign** button.

Depending on your modifications, the new distribution points are added to the list or existing distribution points are removed from the list.

Enabling a push server

In Open Single Management Platform, a distribution point can work as a push server for the devices managed through the mobile protocol and for the devices managed by Network Agent. For example, a push server must be enabled if you want to be able to force synchronization of KasperskyOS devices with Administration Server. A push server has the same scope of managed devices as the distribution point on which the push server is enabled. If you have several distribution points assigned for the same administration group, you can enable push server on each of the distribution points. In this case, Administration Server balances the load between the distribution points.

You might want to use distribution points as push servers to make sure that there is continuous connectivity between a managed device and the Administration Server. Continuous connectivity is needed for some operations, such as running and stopping local tasks, receiving statistics for a managed application, or creating a tunnel. If you use a distribution point as a push server, you do not have to use the **Do not disconnect from the Administration Server** option on managed devices or send packets to the UDP port of the Network Agent.

A push server supports the load of up to 50,000 simultaneous connections.

To enable push server on a distribution point:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.
2. On the **General** tab, select the **Distribution points** section.
3. Click the name of the distribution point on which you want to enable the push server.

   The distribution point properties window opens.
4. On the **General** section, enable the **Run push server** option.
5. In the **Push server port** field, type the port number. You can specify number of any unoccupied port.

6. In the **Address for remote hosts** field, specify the IP address or the name of the distribution point device.

7. Click the **OK** button.

The push server is enabled on the selected distribution point.

About device statuses

Open Single Management Platform assigns a status to each managed device. The particular status depends on whether the conditions defined by the user are met. In some cases, when assigning a status to a device, Open Single Management Platform takes into consideration the device's visibility flag on the network (see the table below). If Open Single Management Platform does not find a device on the network within two hours, the visibility flag of the device is set to *Not Visible*.

The statuses are the following:

- *Critical* or *Critical/Visible*
- *Warning* or *Warning/Visible*
- *OK* or *OK/Visible*

The table below lists the default conditions that must be met to assign the *Critical* or *Warning* status to a device, with all possible values.

Conditions for assigning a status to a device

Condition	Condition description	Available values
Security application is not installed	Network Agent is installed on the device, but a security application is not installed.	• Toggle button is on. • Toggle button is off.
Too many viruses detected	Some viruses have been found on the device by a task for virus detection, for example, the Malware scan task, and the number of viruses found exceeds the specified value.	More than 0.
Real-time protection level differs from the level set by the Administrator	The device is visible on the network, but the real-time protection level differs from the level set (in the condition) by the administrator for the device status.	• Stopped. • Paused. • Running.
Malware scan has not been performed in a long time	The device is visible on the network and a security application is installed on the device, but neither the *Malware scan* task nor a local scan task has been run within the specified time interval. The condition is applicable only to devices that were added to the Administration Server database 7 days ago or earlier.	More than 1 day.
Databases are outdated	The device is visible on the network and a security application is installed on the device, but the anti-virus databases have not been updated on this device within the specified time interval. The condition is applicable only to devices that were added to the Administration Server database 1 day ago or earlier.	More than 1 day.
Not connected in a long time	Network Agent is installed on the device, but the device has not connected to an Administration Server within the specified time interval, because the device was turned off.	More than 1 day.
Active threats are detected	The number of unprocessed objects in the **Active threats** folder exceeds the specified value.	More than 0 items.

Condition	Condition description	Available values
Restart is required	The device is visible on the network, but an application requires the device restart longer than the specified time interval and for one of the selected reasons.	More than 0 minutes.
Incompatible applications are installed	The device is visible on the network, but software inventory performed through Network Agent has detected incompatible applications installed on the device.	• Toggle button is off. • Toggle button is on.
Software vulnerabilities have been detected	The device is visible on the network and Network Agent is installed on the device, but the *Find vulnerabilities and required updates* task has detected vulnerabilities with the specified severity level in applications installed on the device.	• Critical. • High. • Medium. • Ignore if the vulnerability cannot be fixed. • Ignore if an update is assigned for installation.
License expired	The device is visible on the network, but the license has expired.	• Toggle button is off. • Toggle button is on.
License expires soon	The device is visible on the network, but the license will expire on the device in less than the specified number of days.	Less than 0 days.
Check for Windows Update updates has not been performed in a long time	The device is visible on the network, but the *Find vulnerabilities and required updates* task has not been run within the specified time interval.	More than 1 day.
Invalid encryption status	Network Agent is installed on the device, but the device encryption result is equal to the specified value.	• Does not comply with the policy due to the user's refusal (for external devices only). • Does not comply with the policy due to an error. • Restart is required when applying the policy. • No encryption policy is specified. • Not supported. • When applying the policy.
Mobile device settings do not comply with the policy	The mobile device settings are other than the settings that were specified in the Kaspersky Endpoint Security for Android policy during the check of compliance rules.	• Toggle button is off. • Toggle button is on.
Unprocessed security issues detected	Some unprocessed security issues have been found on the device. Security issues can be created either automatically, through managed Kaspersky applications installed on the client device, or manually by the administrator.	• Toggle button is off. • Toggle button is on.
Device status defined by application	The status of the device is defined by the managed application.	• Toggle button is off. • Toggle button is on.
Device is out of disk space	Free disk space on the device is less than the specified value, or the device could not be synchronized with the Administration Server. The *Critical* or *Warning* status is changed to the *OK* status when the device is successfully synchronized with the Administration Server and free space on the device is greater than or equal to the specified value.	More than 0 MB.
Device has become unmanaged	During device discovery, the device was recognized as visible on the network, but more than three attempts to synchronize with the Administration Server failed.	• Toggle button is off. • Toggle button is on.
Protection is disabled	The device is visible on the network, but the security application on the device has been disabled for longer than the specified time interval. In this case, the state of the security application is *stopped* or *failure*, and differs from the following: *starting*, *running*, or *suspended*.	More than 0 minutes.
Security application is not running	The device is visible on the network and a security application is installed on the device but is not running.	• Toggle button is off. • Toggle button is on.

Open Single Management Platform allows you to set up automatic switching of the status of a device in an administration group when specified conditions are met. When the specified conditions are met, the client device is assigned one of the following statuses: *Critical* or *Warning*. When the specified conditions are not met, the client device is assigned the *OK* status.

Different statuses may correspond to different values of one condition. For example, by default, if the **Databases are outdated** condition has the **More than 3 days** value, the client device is assigned the *Warning* status; if the value is **More than 7 days**, the *Critical* status is assigned.

> When Open Single Management Platform assigns a status to a device, for some conditions (see the Condition description column in the table above) the visibility flag is taken into consideration. For example, if a managed device was assigned the *Critical* status because the Databases are outdated condition was met, and later the visibility flag was set for the device, then the device is assigned the *OK* status.

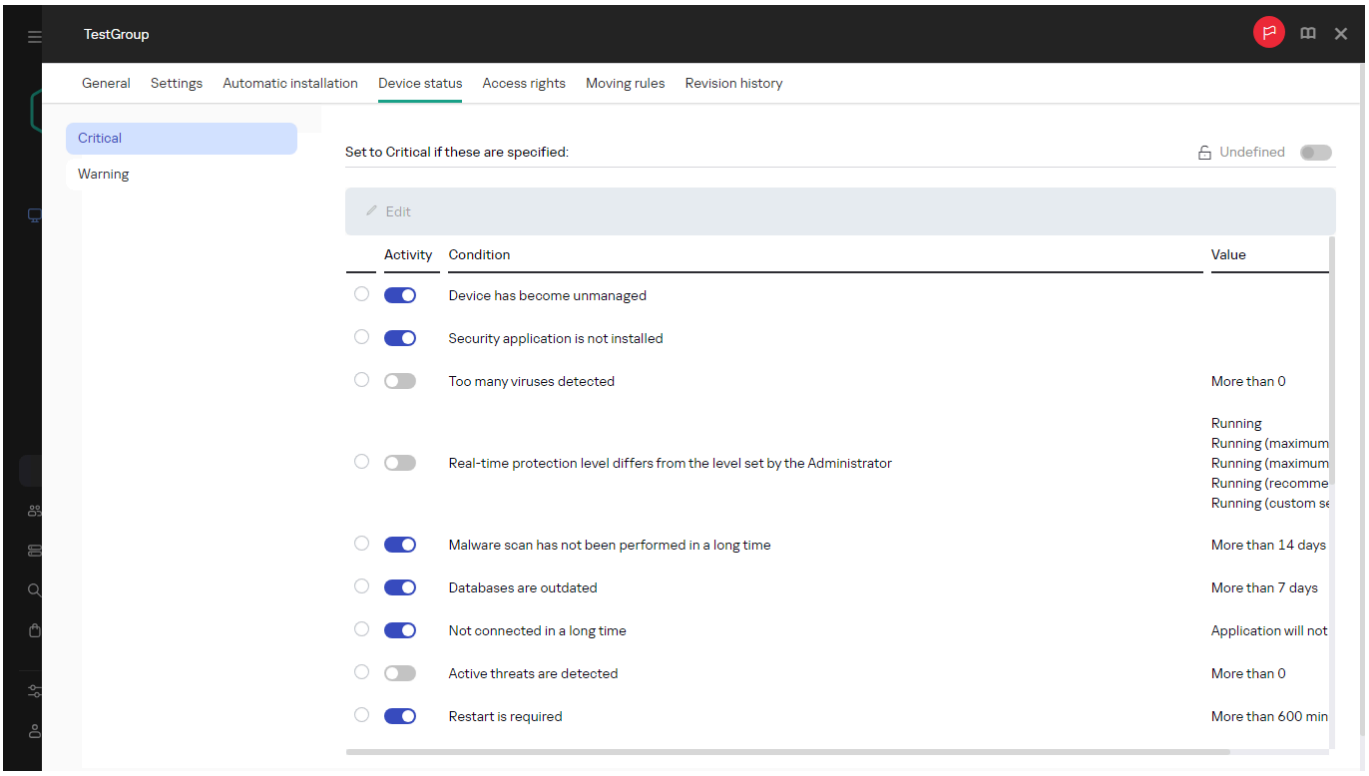
Configuring the switching of device statuses

You can change conditions to assign the *Critical* or *Warning* status to a device.

To enable changing the device status to Critical:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.
2. Go to the properties of the administration group for which you want to change the device statuses in one of the following ways:
   - Click the name of the administration group.
   - Click the **Show hierarchy** button at the top of the page. In the navigation pane that opens, hover the mouse cursor over the name of the required group, and then click the settings icon (⚙).

   The administration group properties window opens.
3. In the properties window, select the **Device status** tab.
4. In the left pane, select **Critical**.
5. In the right pane, in the **Set to Critical if the following settings are specified** section, enable the condition to switch a device to the *Critical* status.



> You can change only settings that are not locked in the parent policy.

6. Select the radio button next to the condition in the list.

7. In the upper-left corner of the list, click the **Edit** button.

8. Set the required value for the selected condition.

   Values cannot be set for every condition.

9. Click **OK**.

When specified conditions are met, the managed device is assigned the *Critical* status.

To enable changing the device status to Warning:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.

2. In the list of groups that opens, click the link with the name of a group for which you want to change switching the device statuses.

3. In the properties window that opens, select the **Device status** tab.

4. In the left pane, select **Warning**.

5. In the right pane, in the **Set to Warning if the following settings are specified** section, enable the condition to switch a device to the *Warning* status.

> You can change only settings that are not locked in the parent policy.

6. Select the radio button next to the condition in the list.

7. In the upper-left corner of the list, click the **Edit** button.

8. Set the required value for the selected condition.

   Values cannot be set for every condition.

9. Click **OK**.

When specified conditions are met, the managed device is assigned the *Warning* status.

Device selections

Device selections are a tool for filtering devices according to specific conditions. You can use device selections to manage several devices: for example, to view a report about only these devices or to move all of these devices to another group.

Open Single Management Platform provides a broad range of *predefined selections* (for example, **Devices with Critical status**, **Protection is disabled**, **Active threats are detected**). Predefined selections cannot be deleted. You can also create and configure additional *user-defined selections*.

In user-defined selections, you can set the search scope and select all devices, managed devices, or unassigned devices. Search parameters are specified in the conditions. In the device selection you can create several conditions with different search parameters. For example, you can create two conditions and specify different IP ranges in each of them. If several conditions are specified, a selection displays the devices that meet any of the conditions. By contrast, search parameters within a condition are superimposed. If both an IP range and the name of an installed application are specified in a condition, only those devices will be displayed where both the application is installed and the IP address belongs to the specified range.

Viewing the device list from a device selection

Open Single Management Platform allows you to view the list of devices from a device selection.

To view the device list from the device selection:

1. In the main menu, go to the **Asset management** → **Device selections** or **Discovery & deployment** → **Device selections** section.

2. In the selection list, click the name of the device selection.

   The page displays a table with information about the devices included in the device selection.

3. You can group and filter the data of the device table as follows:

   - Click the settings icon (⚙), and then select the columns to be displayed in the table.

   - Click the filter icon (▽), and then specify and apply the filter criterion in the invoked menu.

     The filtered table of devices is displayed.

You can select one or several devices in the device selection and click the **New task** button to create a task that will be applied to these devices.

To move the selected devices of the device selection to another administration group, click the **Move to group** button, and then select the target administration group.

Creating a device selection

To create a device selection:

1. In the main menu, go to **Asset management** → **Device selections**.

   A page with a list of device selections is displayed.

2. Click the **Add** button.

   The **Device selection settings** window opens.

3. Enter the name of the new selection.

4. Specify the group that contains the devices to be included in the device selection:

   - **Find any devices**—Searching for devices that meet the selection criteria and included in the **Managed Devices** or **Unassigned devices** group.

   - **Find managed devices**—Searching for devices that meet the selection criteria and included in the **Managed Devices** group.

   - **Find unassigned devices**—Searching for devices that meet the selection criteria and included in the **Unassigned devices** group.

   You can enable the **Include data from secondary Administration Servers** check box to enable searching for devices that meet the selection criteria and managed by secondary Administration Servers.

5. Click the **Add** button.

6. In the window that opens, specify conditions that must be met for including devices in this selection, and then click the **OK** button.

7. Click the **Save** button.

The device selection is created and added to the list of device selections.

Configuring a device selection

To configure a device selection:

1. In the main menu, go to **Asset management** → **Device selections**.

   A page with a list of device selections is displayed.

2. Select the relevant user-defined device selection, and click the **Properties** button.

   The **Device selection settings** window opens.

3. On the **General** tab, click the **New condition** link.

4. Specify conditions that must be met for including devices in this selection.

5. Click the **Save** button.

   The settings are applied and saved.

Below are descriptions of the conditions for assigning devices to a selection. Conditions are combined by using the OR logical operator: the selection will contain devices that comply with at least one of the listed conditions.

General

In the **General** section, you can change the name of the selection condition and specify whether that condition must be inverted:

Invert selection condition

By default, this option is disabled.

If you enable this option, the specified selection condition will be inverted. The selection will include all devices that do not meet the condition.

The following example shows the logic of the inversion work:

- **Condition**: On your device, there is an operating system with the Linux type and the 00.0.000 build number.
- **Inverted condition**: On your device, there is an operating system with NOT (the Linux type and the 00.0.000 build number), which is equivalent to the following:

  There is an operating system with the type NOT Linux OR build number NOT 00.0.000.

Thus, the selection will include the devices with an operating system, for example, of the Windows type and the 00.0.000 build number, or the Linux type and any build number other than 00.0.000.

Network infrastructure

In the **Network** subsection, you can specify the criteria that will be used to include devices in the selection according to their network data:

- **Device name**

  Windows network name (NetBIOS name) of the device, or the IPv4 or IPv6 address.

- **Domain**

  Displays all devices included in the specified workgroup.

- **Administration group**

  Displays devices included in the specified administration group.

- **Description**

  Text in the device properties window: in the **Description** field of the **General** section.

  To describe text in the **Description** field, you can use the following characters:

  - Within a word:

    - *****. Replaces any string with any number of characters.

    Example:

    To describe words such as **Server** or **Server's**, you can enter **Server***.

    - **?.** Replaces any single character.

    Example:

    To describe phrases such as **SUSE Linux Enterprise Server 12** or **SUSE Linux Enterprise Server 15**, you can enter **SUSE Linux Enterprise Server 1?**.

    Asterisk (*****) or question mark (**?**) cannot be used as the first character in the query.

  - To find several words:

    - Space. Displays all the devices whose descriptions contain any of the listed words.

    Example:

    To find a phrase that contains **Secondary** or **Virtual** words, you can include **Secondary Virtual** line in your query.

    - **+**. When a plus sign precedes a word, all search results will contain this word.

    Example:

    To find a phrase that contains both **Secondary** and **Virtual**, enter the **+Secondary+Virtual** query.

    - **-**. When a minus sign precedes a word, no search results will contain this word.

    Example:

    To find a phrase that contains **Secondary** and does not contain **Virtual**, enter the **+Secondary-Virtual** query.

    - **"<some text>"**. Text enclosed in quotation marks must be present in the text.

    Example:

    To find a phrase that contains **Secondary Server** word combination, you can enter **"Secondary Server"** in the query.

- **IP range**

  If this option is enabled, you can enter the initial and final IP addresses of the IP range in which the relevant devices must be included.

  By default, this option is disabled.

- **Managed by a different Administration Server**

  Select one of the following values:

  - **Yes**. A device moving rule only applies to client devices managed by other Administration Servers. These Servers are different from the Server on which you configure the device moving rule.
  - **No**. The device moving rule only applies to client devices managed by the current Administration Server.
  - **No value is selected**. The condition does not apply.

In the **Domain controller** subsection, you can configure criteria for including devices into a selection based on domain membership:

- **Device is in a domain organizational unit**

  If this option is enabled, the selection includes devices from the domain organizational unit specified in the entry field.

  By default, this option is disabled.

- **This device is a member of the domain security group**

  If this option is enabled, the selection includes devices from the domain security group specified in the entry field.

  By default, this option is disabled.

In the **Network activity** subsection, you can specify the criteria that will be used to include devices in the selection according to their network activity:

- **Acts as a distribution point**

  In the drop-down list, you can set up the criterion for including devices in the selection when performing search:

  - **Yes**. The selection includes devices that act as distribution points.
  - **No**. Devices that act as distribution points are not included in the selection.
  - **No value is selected**. The criterion will not be applied.

- **Do not disconnect from the Administration Server**

  In the drop-down list, you can set up the criterion for including devices in the selection when performing search:

  - **Enabled**. The selection will include devices on which the **Do not disconnect from the Administration Server** check box is selected.
  - **Disabled**. The selection will include devices on which the **Do not disconnect from the Administration Server** check box is cleared.
  - **No value is selected**. The criterion will not be applied.

- **Connection profile switched**

  In the drop-down list, you can set up the criterion for including devices in the selection when performing search:

  - **Yes**. The selection will include devices that connected to the Administration Server after the connection profile was switched.
  - **No**. The selection will not include devices that connected to the Administration Server after the connection profile was switched.
  - **No value is selected**. The criterion will not be applied.

- **Last connected to Administration Server**

  You can use this check box to set a search criterion for devices according to the time they last connected to the Administration Server.

  If this check box is selected, in the entry fields you can specify the time interval (date and time) during which the last connection was established between Network Agent installed on the client device and the Administration Server. The selection will include devices that fall within the specified interval.

  If this check box is cleared, the criterion will not be applied.

  By default, this check box is cleared.

- **New devices detected by network poll**

  Searches for new devices that have been detected by network polling over the last few days.

  If this option is enabled, the selection only includes new devices that have been detected by device discovery over the number of days specified in the **Detection period (days)** field.

  If this option is disabled, the selection includes all devices that have been detected by device discovery.

  By default, this option is disabled.

- **Device is visible**

  In the drop-down list, you can set up the criterion for including devices in the selection when performing search:

  - **Yes**. The application includes in the selection devices that are currently visible in the network.
  - **No**. The application includes in the selection devices that are currently invisible in the network.
  - **No value is selected**. The criterion will not be applied.

Device statuses

In the **Managed device status** subsection, you can configure criteria for including devices into a selection based on the description of the devices status from a managed application:

- **Device status**

  Drop-down list in which you can select one of the device statuses: *OK*, *Critical*, or *Warning*.

- **Real-time protection status**

  Drop-down list, in which you can select the real-time protection status. Devices with the specified real-time protection status are included in the selection.

- **Device status description**

  In this field, you can select the check boxes next to conditions that, if met, assign one of the following statuses to the device: *OK, Critical*, or *Warning*.

In the **Status of components in managed applications** subsection, you can configure criteria for including devices in a selection according to the statuses of components in managed applications:

- **Data Leakage Prevention status**

  Search for devices by the status of Data Leakage Prevention (*Unknown, Stopped, Starting, Paused, Running, Failed*).

- **Collaboration servers protection status**

  Search for devices by the status of server collaboration protection (*Unknown*, *Stopped*, *Starting, Paused*, *Running*, *Failed*).

- **Anti-virus protection status of mail servers**

  Search for devices by the status of Mail Server protection (*Unknown, Stopped, Starting, Paused, Running, Failed*).

- **Endpoint Sensor status**

  Search for devices by the status of the Endpoint Sensor component (*Unknown, Stopped, Starting, Paused, Running, Failed*).

In the **Status-affecting problems in managed applications** subsection, you can specify the criteria that will be used to include devices in the selection according to the list of possible problems detected by a managed application. If at least one problem that you select exists on a device, the device will be included in the selection. When you select a problem listed for several applications, you have the option to select this problem in all of the lists automatically.

You can select check boxes for descriptions of statuses from the managed application; upon receipt of these statuses, the devices will be included in the selection. When you select a status listed for several applications, you have the option to select this status in all of the lists automatically.

System details

In the **OS** section, you can specify the criteria that will be used to include devices in the selection according to their operating system type.

- **Platform type**

  If the check box is selected, you can select an operating system from the list. Devices with the specified operating systems installed are included in the search results.

- **OS service pack version**

  In this field, you can specify the package version of the operating system (in the *X.Y* format), which will determine how the moving rule is applied to the device. By default, no version value is specified.

- **OS architecture**

  In the drop-down list, you can select the architecture for the operating system, which will determine how the moving rule is applied to the device (**Unknown**, **x86**, **AMD64**, or **IA64**). By default, no option is selected in the list so that the operating system's architecture is not defined.

- **OS build**

  This setting is applicable to Windows operating systems only.

  The build number of the operating system. You can specify whether the selected operating system must have an equal, earlier, or later build number. You can also configure searching for all build numbers except the specified one.

- **OS release number**

  This setting is applicable to Windows operating systems only.

  The release identifier (ID) of the operating system. You can specify whether the selected operating system must have an equal, earlier, or later release ID. You can also configure searching for all release ID numbers except the specified one.

In the **Virtual machines** section, you can set up the criteria to include devices in the selection according to whether these are virtual machines or part of virtual desktop infrastructure (VDI):

- **This is a virtual machine**

  In the drop-down list, you can select the following options:

  - **Undefined**.
  - **No**. Find devices that are not virtual machines.
  - **Yes**. Find devices that are virtual machines.

- **Virtual machine type**

  In the drop-down list, you can select the virtual machine manufacturer.

  This drop-down list is available if the **Yes** or **Not important** value is selected in the **This is a virtual machine** drop-down list.

- **Part of Virtual Desktop Infrastructure**

  In the drop-down list, you can select the following options:

  - **Undefined**.
  - **No**. Find devices that are not part of Virtual Desktop Infrastructure.
  - **Yes**. Find devices that are part of the Virtual Desktop Infrastructure (VDI).

In the **Hardware registry** subsection, you can configure criteria for including devices into a selection based on their installed hardware:

Ensure that the lshw utility is installed on Linux devices from which you want to fetch hardware details. Hardware details fetched from virtual machines may be incomplete depending on the hypervisor used.

- **Device**

  In the drop-down list, you can select a unit type. All devices with this unit are included in the search results.

  The field supports the full-text search.

- **Vendor**

  In the drop-down list, you can select the name of a unit manufacturer. All devices with this unit are included in the search results.

  The field supports the full-text search.

- **Device name**

  The device with the specified name is included in the selection.

- **Description**

  Description of the device or hardware unit. Devices with the description specified in this field are included in the selection.

  A device's description in any format can be entered in the properties window of that device. The field supports the full-text search.

- **Device vendor**

  Name of the device manufacturer. Devices produced by the manufacturer specified in this field are included in the selection.

  You can enter the manufacturer's name in the properties window of a device.

- **Serial number**

  All hardware units with the serial number specified in this field will be included in the selection.

- **Inventory number**

  Equipment with the inventory number specified in this field will be included in the selection.

- **User**

  All hardware units of the user specified in this field will be included in the selection.

- **Location**

  Location of the device or hardware unit (for example, at the HQ or a branch office). Computers or other devices that are deployed at the location specified in this field will be included in the selection.

  You can describe the location of a device in any format in the properties window of that device.

- **CPU clock rate, in MHz, from**

  The minimum clock rate of a CPU. Devices with a CPU that matches the clock rate range specified in the entry fields (inclusive) will be included in the selection.

- **CPU clock rate, in MHz, to**

  The maximum clock rate of a CPU. Devices with a CPU that matches the clock rate range specified in the entry fields (inclusive) will be included in the selection.

- **Number of virtual CPU cores, from**

  The minimum number of virtual CPU cores. Devices with a CPU that matches the range of the virtual cores number specified in the entry fields (inclusive) will be included in the selection.

- **Number of virtual CPU cores, to**

  The maximum number of virtual CPU cores. Devices with a CPU that matches the range of the virtual cores number specified in the entry fields (inclusive) will be included in the selection.

- **Hard drive volume, in GB, from**

  The minimum volume of the hard drive on the device. Devices with a hard drive that matches the volume range specified in the entry fields (inclusive) will be included in the selection.

- **Hard drive volume, in GB, to**

  The maximum volume of the hard drive on the device. Devices with a hard drive that matches the volume range specified in the entry fields (inclusive) will be included in the selection.

- **RAM size, in MB, from**

  The minimum size of the device RAM. Devices with RAM that matches the size range specified in the entry fields (inclusive) will be included in the selection.

- **RAM size, in MB, to**

  The maximum size of the device RAM. Devices with RAM that matches the size range specified in the entry fields (inclusive) will be included in the selection.

Third-party software details

In the **Applications registry** subsection, you can set up the criteria to search for devices according to applications installed on them:

- **Application name**

  Drop-down list in which you can select an application. Devices on which the specified application is installed, are included in the selection.

- **Application version**

  Entry field in which you can specify the version of selected application.

- **Vendor**

  Drop-down list in which you can select the manufacturer of an application installed on the device.

- **Application status**

  A drop-down list in which you can select the status of an application (*Installed*, *Not installed*). Devices on which the specified application is installed or not installed, depending on the selected status, will be included in the selection.

- **Find by update**

  If this option is enabled, search will be performed using the details of updates for applications installed on the relevant devices. After you select the check box, the **Application name**, **Application version**, and **Application status** fields change to **Update name**, **Update version**, and **Status** respectively.

  By default, this option is disabled.

- **Name of incompatible security application**

  Drop-down list in which you can select third-party security applications. During the search, devices on which the specified application is installed, are included in the selection.

- **Application tag**

  In the drop-down list, you can select the application tag. All devices that have installed applications with the selected tag in the description are included in the device selection.

- **Apply to devices without the specified tags**

  If this option is enabled, the selection includes devices with descriptions that contain none of the selected tags.

  If this option is disabled, the criterion is not applied.

  By default, this option is disabled.

In the **Vulnerabilities and updates** subsection, you can specify the criteria that will be used to include devices in the selection according to their Windows Update source:

WUA is switched to Administration Server

You can select one of the following search options from the drop-down list:

- **Yes**. If this option is selected, the search results will include devices that receive updates through Windows Update from the Administration Server.

- **No**. If this option is selected, the results will include devices that receive updates through Windows Update from another sources.

Details of Kaspersky applications

In the **Kaspersky applications** subsection, you can configure criteria for including devices in a selection based on the selected managed application:

- **Application name**

  In the drop-down list, you can set a criterion for including devices in a selection when search is performed by the name of a Kaspersky application.

  The list provides only the names of applications with management plug-ins installed on the administrator's workstation.

If no application is selected, the criterion will not be applied.

- **Application version**

  In the entry field, you can set a criterion for including devices in a selection when search is performed by the version number of a Kaspersky application.

  If no version number is specified, the criterion will not be applied.

- **Critical update name**

  In the entry field, you can set a criterion for including devices in a selection when search is performed by application name or by update package number.

  If the field is left blank, the criterion will not be applied.

- **Application status**

  A drop-down list in which you can select the status of an application (*Installed*, *Not installed*). Devices on which the specified application is installed or not installed, depending on the selected status, will be included in the selection.

- **Select the period of the last update of modules**

  You can use this option to set a criterion for searching devices by time of the last update of modules of applications installed on those devices.

  If this check box is selected, in the entry fields you can specify the time interval (date and time) during which the last update of modules of applications installed on those devices was performed.

  If this check box is cleared, the criterion will not be applied.

  By default, this check box is cleared.

- **Device is managed through Administration Server**

  In the drop-down list, you can include in the selection the devices managed through Open Single Management Platform:

  - **Yes**. The application includes in the selection devices managed through Open Single Management Platform.
  - **No**. The application includes devices in the selection if they are not managed through Open Single Management Platform.
  - **No value is selected**. The criterion will not be applied.

- **Security application is installed**

  In the drop-down list, you can include in the selection all devices with the security application installed:

  - **Yes**. The application includes in the selection all devices with the security application installed.
  - **No**. The application includes in the selection all devices with no security application installed.
  - **No value is selected**. The criterion will not be applied.

In the **Anti-virus protection** subsection, you can set up the criteria for including devices in a selection based on their protection status:

- **Databases released**

  If this option is selected, you can search for client devices by anti-virus database release date. In the entry fields you can set the time interval, on the basis of which the search is performed.

  By default, this option is disabled.

- **Database records count**

  If this option is enabled, you can search for client devices by number of database records. In the entry fields you can set the lower and upper threshold values for anti-virus database records.

  By default, this option is disabled.

- **Last scanned**

  If this check option is enabled, you can search for client devices by time of the last malware scan. In the entry fields you can specify the time period within which the last malware scan was performed.

  By default, this option is disabled.

- **Threats detected**

  If this option is enabled, you can search for client devices by number of viruses detected. In the entry fields you can set the lower and upper threshold values for the number of viruses found.

  By default, this option is disabled.

In the **Encryption** subsection, you can configure the criterion for including devices in a selection based on the selected encryption algorithm:

Encryption algorithm

Advanced Encryption Standard (AES) symmetrical block cipher algorithm. In the drop-down list, you can select the encryption key size (56-bit, 128-bit, 192-bit, or 256-bit).

Available values: *AES56*, *AES128*, *AES192*, and *AES256*.

The **Application components** subsection contains the list of components of those applications that have corresponding management plug-ins installed in OSMP Console.

In the **Application components** subsection, you can specify criteria for including devices in a selection according to the statuses and version numbers of the components that refer to the application that you select:

- **Status**

  Search for devices according to the component status sent by an application to the Administration Server. You can select one of the following statuses: *N/A, Stopped*, *Paused*, *Starting*, *Running, Failed, Not installed, Not supported by license*. If the selected component of the application installed on a managed device has the specified status, the device is included in the device selection.

  Statuses sent by applications:

  - *Stopped*—The component is disabled and not working at the moment.
  - *Paused*—The component is suspended, for example, after the user has paused protection in the managed application.
  - *Starting*—The component is currently in the process of initialization.
  - *Running*—The component is enabled and working properly.
  - *Failed*—An error has occurred during the component operation.
  - *Not installed*—The user did not select the component for installation when configuring custom installation of the application.
  - *Not supported by license*—The license does not cover the selected component.

  Unlike other statuses, the *N/A* status is not sent by applications. This option shows that the applications have no information about the selected component status. For example, this can happen when the selected component does not belong to any of the applications installed on the device, or when the device is turned off.

- **Version**

  Search for devices according to the version number of the component that you select in the list. You can type a version number, for example `3.4.1.0`, and then specify whether the selected component must have an equal, earlier, or later version. You can also configure searching for all versions except the specified one.

Tags

In the **Tags** section, you can configure criteria for including devices into a selection based on key words (tags) that were previously added to the descriptions of managed devices:

Apply if at least one specified tag matches

If this option is enabled, the search results will show devices with descriptions that contain at least one of the selected tags.

If this option is disabled, the search results will only show devices with descriptions that contain all the selected tags.

By default, this option is disabled.

To add tags to the criterion, click the **Add** button, and select tags by clicking the **Tag** entry field. Specify whether to include or exclude the devices with the selected tags in the device selection.

- **All devices that have this tag**

  If this option is selected, the search results will display the devices whose descriptions contain the selected tag. To find devices, you can use the asterisk, which stands for any string with any number of characters.

  By default, this option is selected.

- **All devices that do not have this tag**

  If this option is selected, the search results will display the devices whose descriptions do not contain the selected tag. To find devices, you can use the asterisk, which stands for any string with any number of characters.

Users

In the **Users** section, you can set up the criteria to include devices in the selection according to the accounts of users who have logged in to the operating system.

- **Last user who logged in to the system**

  If this option is enabled, you can select the user account for configuring the criterion. The search results include devices on which the selected user performed the last login to the system.

- **User who logged in to the system at least once**

  If this option is enabled, click the **Browse** button to specify a user account. The search results include devices on which the specified user logged in to the system at least once.

Device owner

In the **Device owner** section, you can set up the criteria to include devices in the selection according to the registered owners of the device, their roles, and their membership in security groups:

- **Device owner**

  Select a device owner's user name from an internal security group. Learn more about users and user roles in this section.

  No more than one user can be registered as the device owner.

- **Device owner's membership in Active Directory security group**

  Select an external Active Directory security group that the device owner belongs to.

  The user can be a part of an Active Directory security group or a part of a group that is included in this Active Directory security group.

- **Device owner's role**

  Select the device owner's assigned role. Learn more about user roles in this article.

- **Device owner's membership in an internal security group**

  Select an internal security group that the device owner belongs to.

Exporting the device list from a device selection

Open Single Management Platform allows you to save information about devices from a device selection and export it as a CSV or a TXT file.

To export the device list from the device selection:

1. Open the table with the devices from the device selection.
2. Use one of the following ways to select the devices that you want to export:
   - To select particular devices, select the check boxes next to them.
   - To select all devices from the current table page, select the check box in the device table header, and then select the **Select all on current page** check box.
   - To select all devices from the table, select the check box in the device table header, and then select the **Select all** check box.
3. Click the **Export to CSV** or **Export to TXT** button. All information about the selected devices included in the table will be exported.

Note that if you applied a filter criterion to the device table, only the filtered data from the displayed columns will be exported.

Removing devices from administration groups in a selection

When working with a device selection, you can remove devices from administration groups right in this selection, without switching to the administration groups from which these devices must be removed.

To remove devices from administration groups:

1. In the main menu, go to **Asset management** → **Device selections** or **Discovery & deployment** → **Device selections**.
2. In the selection list, click the name of the device selection.

   The page displays a table with information about the devices included in the device selection.
3. Select the devices that you want to remove, and then click **Delete**.

   The selected devices are removed from their respective administration groups.

Device tags

This section describes device tags, and provides instructions for creating and modifying them as well as for tagging devices manually or automatically.

Device tags

Open Single Management Platform allows you to *tag* devices. A tag is the string value that can be used for grouping, describing, or finding devices. Tags assigned to devices can be used for creating selections, for finding devices, and for distributing devices among administration groups.

You can tag devices manually or automatically. If you want to tag an individual device, you can use manual tagging. Auto-tagging is performed by Open Single Management Platform in one of the following ways:

- In accordance with the specified tagging rules.
- By an application.

> We do not recommend that you use different ways of tagging to assign the same tag. For example, if the tag is assigned by the rule, it is not recommended to manually assign this tag to devices.

If the tags are assigned by rules, devices are tagged automatically when the specified rules are met. An individual rule corresponds to each tag. Rules are applied to the network properties of the device, operating system, applications installed on the device, and other device properties. For example, you can set up a rule that will assign the `[CentOS]` tag to all devices running CentOS operating system. Then, you can use this tag when creating a device selection; this will help you sort all CentOS devices and assign them a task.

> You can add only one auto-tagging rule per tag. If you create a new rule for the same tag, the previous one will be replaced.

A tag is automatically removed from a device in the following cases:

- When the device stops meeting conditions of the rule that assigns the tag.
- When the rule that assigns the tag is disabled or deleted.

The list of tags and the list of rules on each Administration Server are independent of all other Administration Servers, including a primary Administration Server or subordinate virtual Administration Servers. A rule is applied only to devices from the same Administration Server on which the rule is created.

Creating a device tag

To create a device tag:

1. In the main menu, go to **Asset management** → **Tags** → **Device tags**.
2. Click **Add**.

   A new tag window opens.
3. In the **Tag** field, enter the tag name.
4. Click **Save** to save the changes.

The new tag appears in the list of device tags.

Renaming a device tag

To rename a device tag:

1. In the main menu, go to **Asset management** → **Tags** → **Device tags**.
2. Click the name of the tag that you want to rename.
   A tag properties window opens.
3. In the **Tag** field, change the tag name.
4. Click **Save** to save the changes.

The updated tag appears in the list of device tags.

Deleting a device tag

You can delete only manually assigned tags.

To delete a manually assigned device tag:

1. In the main menu, go to **Asset management** → **Tags** → **Device tags**.
   The list of tags is displayed.
2. Select the device tag that you want to delete.
3. Click the **Delete** button.
4. In the window that opens, click **Yes**.

The device tag is deleted. The deleted tag is automatically removed from all of the devices to which it was assigned.

When you delete a tag assigned to the device by an auto-tagging rule, the rule is not deleted, and the tag will be assigned to a new device when the device first meets the rule conditions. If you delete an auto-tagging rule, the tag specified in the rule conditions will be removed from all devices to which it was assigned but will not be deleted from the list of tags. If necessary, you can manually delete the tag from the list.

The deleted tag is not removed automatically from the device if this tag is assigned to the device by an application or Network Agent. To manage such tags, use the klscflag utility.

Viewing devices to which a tag is assigned

To view devices to which a tag is assigned:

1. In the main menu, go to **Asset management** → **Tags** → **Device tags**.
2. Click the **View devices** link next to the tag for which you want to view assigned devices.

   You will be redirected to the **Assets (Devices)** section of the main menu, with the devices filtered by the tag for which you clicked the **View devices** link.
3. If you want to return to the list of device tags, click the **Back** button of your browser.

After you view the devices to which the tag is assigned, you can either create and assign a new tag or assign the existing tag to other devices. In this case, you have to remove the filter by tag, select the devices, and then assign the tag.

Viewing tags assigned to a device

To view tags assigned to a device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.
2. Click the name of the device whose tags you want to view.
3. In the device properties window that opens, select the **Tags** tab.

   The list of tags assigned to the selected device is displayed. In the **Tag assigned** column you can view how the tag was assigned.

In the toolbar, you can do the following:

- Assign another tag to the device.
- Remove an already assigned tag.
- View all device tags that exist on the Administration Server.

> You cannot view the assigned tags locally on the device.

Tagging a device manually

To assign a tag to a device manually:

1. View tags assigned to the device to which you want to assign another tag.
2. Click **Add**.

3. In the window that opens, do one of the following:
   - To create and assign a new tag, select **Create new tag**, and then specify the name of the new tag.
   - To select an existing tag, select **Assign existing tag**, and then select the necessary tag in the drop-down list.
4. Click **OK** to apply the changes.
5. Click **Save** to save the changes.

   The selected tag is assigned to the device.

Removing an assigned tag from a device

To remove a tag from a device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.
2. Click the name of the device whose tags you want to view.
3. In the device properties window that opens, select the **Tags** tab.
4. Select the check box next to the tag that you want to remove.
5. At the top of the list, click the **Unassign tag** button.
6. In the window that opens, click **Yes**.

   The tag is removed from the device.

The unassigned device tag is not deleted. If you want, you can delete it manually.

You cannot manually remove tags assigned to the device by applications or Network Agent. To manage these tags, use the klscflag utility.

Viewing rules for tagging devices automatically

To view rules for tagging devices automatically,

Do any of the following:

- In the main menu, go to **Asset management** → **Tags** → **Auto-tagging rules**.
- In the main menu, go to **Asset management** → **Tags** → **Device tags**, and then click the **Set up auto-tagging rules** link.
- View tags assigned to a device and then click the **Settings** button.

The list of rules for auto-tagging devices appears.

Editing a rule for tagging devices automatically

To edit a rule for tagging devices automatically:

1. View rules for tagging devices automatically.
2. Click the name of the rule that you want to edit.

   A rule settings window opens.
3. Edit the general properties of the rule:

   a. In the **Rule name** field, change the rule name.

      The name cannot contain more than 256 characters.

   b. Do any of the following:

      - Enable the rule by switching the toggle button to **Rule enabled**.
      - Disable the rule by switching the toggle button to **Rule disabled**.
4. Do any of the following:

   - If you want to add a new condition, click the **Add** button, and specify the settings of the new condition in the window that opens.
   - If you want to edit an existing condition, click the name of the condition that you want to edit, and then edit the condition settings.
   - If you want to delete a condition, select the check box next to the name of the condition that you want to delete, and then click **Delete**.
5. Click **OK** in the conditions settings window.
6. Click **Save** to save the changes.

The edited rule is shown in the list.

Creating a rule for tagging devices automatically

To create a rule for tagging devices automatically:

1. View rules for tagging devices automatically.
2. Click **Add**.

   A new rule settings window opens.

3. Configure the general properties of the rule:

   a. In the **Rule name** field, enter the rule name.

      The name cannot be more than 256 characters long.

   b. Do one of the following:

      - Enable the rule by switching the toggle button to **Rule enabled**.
      - Disable the rule by switching the toggle button to **Rule disabled**.

   c. In the **Tag** field, enter the new device tag name or select one of the existing device tags from the list.

      The name cannot be more than 256 characters long.

4. In the conditions section, click the **Add** button to add a new condition.

   A new condition settings window open.

5. Enter the condition name.

   The name cannot be more than 256 characters long. The name must be unique within a rule.

6. Set up the triggering of the rule according to the following conditions. You can select multiple conditions.

   - **Network**—Network properties of the device, such as DNS name of the device or device inclusion in an IP subnet.

     > If case sensitive collation is set for the database that you use for Open Single Management Platform, keep case when you specify a device DNS name. Otherwise, the auto-tagging rule will not work.

   - **Applications**—Presence of Network Agent on the device, operating system type, version, architecture, and package manager.
   - **Virtual machines**—Device belongs to a specific type of virtual machine.
   - **Applications registry**—Presence of applications of different vendors on the device.

7. Click **OK** to save the changes.

   If necessary, you can set multiple conditions for a single rule. In this case, the tag will be assigned to a device if it meets at least one condition.

8. Click **Save** to save the changes.

The newly created rule is enforced on devices managed by the selected Administration Server. If the settings of a device meet the rule conditions, the device is assigned the tag.

Later, the rule is applied in the following cases:

- Automatically and periodically, depending on the server workload
- After you edit the rule

- When you run the rule manually
- After Administration Server detects a change in the settings of a device that meets the rule conditions or the settings of a group that contains such a device

You can create multiple tagging rules. A single device can be assigned multiple tags if you have created multiple tagging rules and if the respective conditions of these rules are met simultaneously. You can view the list of all assigned tags in the device properties.

Running rules for auto-tagging devices

When a rule is run, the tag specified in properties of this rule is assigned to devices that meet conditions specified in properties of the same rule. You can run only active rules.

To run rules for auto-tagging devices:

1. View rules for tagging devices automatically.
2. Select check boxes next to active rules that you want to run.
3. Click the **Run rule** button.

   The selected rules are run.

Deleting a rule for tagging devices automatically

To delete a rule for tagging devices automatically:

1. View rules for tagging devices automatically.
2. Select the check box next to the rule that you want to delete.
3. Click **Delete**.
4. In the window that opens, click **Delete** again.

   The selected rule is deleted. The tag that was specified in properties of this rule is unassigned from all of the devices that it was assigned to.

   The unassigned device tag is not deleted. If you want, you can delete it manually.

Data encryption and protection

Data encryption reduces the risk of unintentional leakage of sensitive and corporate data if your laptop or hard drive is stolen or lost. Also, data encryption allows you to prevent access by unauthorized users and applications.

You can use the data encryption feature if your network includes Windows-based managed devices with Kaspersky Endpoint Security for Windows installed. In this case, on devices running a Windows operating system, you can manage the following types of encryption:

- BitLocker Drive Encryption
- Kaspersky Disk Encryption

We do not recommend enabling encryption in VDI.

By using these components of Kaspersky Endpoint Security for Windows, you can, for example, enable or disable encryption, view the list of encrypted drives, or generate and view reports about encryption.

To configure encryption, define the Kaspersky Endpoint Security for Windows policy in Open Single Management Platform. Kaspersky Endpoint Security for Windows performs encryption and decryption according to the active policy. For detailed instructions on how to configure rules and for a description of encryption features, see the Kaspersky Endpoint Security for Windows Help.

Encryption management for a hierarchy of Administration Servers is currently not available in the Web Console. Use the primary Administration Server to manage encrypted devices.

You can show or hide some of the interface elements related to the encryption management feature by using the user interface settings.

Viewing the list of encrypted drives

In Open Single Management Platform, you can view details about encrypted drives and devices that are encrypted at the drive level. After the information on a drive is decrypted, the drive is automatically removed from the list.

To view the list of encrypted drives,

In the main menu, go to **Operations** → **Data encryption and protection** → **Encrypted drives**.

If the section is not on the menu, this means that it is hidden. In the user interface settings, enable the **Show data encryption and protection** option to display the section.

You can export the list of encrypted drives to a CSV or TXT file. To do this, click the **Export to CSV** or **Export to TXT** button.

Viewing the list of encryption events

When running data encryption or decryption tasks on devices, Kaspersky Endpoint Security for Windows sends Open Single Management Platform information about events of the following types:

- Cannot encrypt or decrypt a file, or create an encrypted archive, due to a lack of free disk space.
- Cannot encrypt or decrypt a file, or create an encrypted archive, due to license issues.

- Cannot encrypt or decrypt a file, or create an encrypted archive, due to missing access rights.
- The application has been prohibited from accessing an encrypted file.
- Unknown errors.

To view a list of events that occurred during data encryption on devices,

In the main menu, go to **Operations** → **Data encryption and protection** → **Encryption events**.

If the section is not on the menu, this means that it is hidden. In the user interface settings, enable the **Show data encryption and protection** option to display the section.

You can export the list of encrypted drives to a CSV or TXT file. To do this, click the **Export to CSV** or **Export to TXT** button.

Alternatively, you can examine the list of encryption events for every managed device.

To view the encryption events for a managed device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.
2. Click on the name of a managed device.
3. On the **General** tab, go to the **Protection** section.
4. Click the **View data encryption errors** link.

Creating and viewing encryption reports

You can generate the following reports:

- Report on encryption status of managed devices. This report provides details about the data encryption of various managed devices. For example, the report shows the number of devices to which the policy with configured encryption rules applies. Also, you can find out, for instance, how many devices need to be rebooted. The report also contains information about the encryption technology and algorithm for every device.
- Report on encryption status of mass storage devices. This report contains similar information as the report on the encryption status of managed devices, but it provides data only for mass storage devices and removable drives.
- Report on rights to access encrypted drives. This report shows which user accounts have access to encrypted drives.
- Report on file encryption errors. This report contains information about errors that occurred when the data encryption or decryption tasks were run on devices.
- Report on blockage of access to encrypted files. This report contains information about blocking application access to encrypted files. This report is helpful if an unauthorized user or application tries to access encrypted files or drives.

You can generate any report in the **Monitoring** → **Reports** section. Alternatively, in the **Operations** → **Data encryption and protection** section, you can generate the following encryption reports:

- Report on encryption status of mass storage devices
- Report on rights to access encrypted drives
- Report on file encryption errors

*To generate an encryption report in the **Data encryption and protection** section:*

1. Make sure that you enabled the **Show data encryption and protection** option in the Interface options.
2. In the policy properties, open the **Event configuration** tab.
3. In the **Critical** section, click **Add event** and select check box next to the event *Error applying file encryption / decryption rules*.
4. Click **OK**.
5. In the main menu, go to **Operations** → **Data encryption and protection**.
6. Open one of the following sections:
   - **Encrypted drives** generates the report on encryption status of mass storage devices or the report on rights to access encrypted drives.
   - **Encryption events** generates the report on file encryption errors.
7. Click the name of the report that you want to generate.

The report generation starts.

Granting access to an encrypted drive in offline mode

A user can request access to an encrypted device, for example, when Kaspersky Endpoint Security is not installed on the managed device. After you receive the request, you can create an access key file and send it to the user. All of the use cases and detailed instructions are provided in the Kaspersky Endpoint Security Help for the required platform.

To grant access to an encrypted drive in offline mode in Kaspersky Endpoint Security for Windows:

1. Get a request access file from a user (a file with the FDERTC extension). Follow the instructions in the Kaspersky Endpoint Security for Windows Help to generate the file in Kaspersky Endpoint Security for Windows.
2. In the main menu, go to **Operations** → **Data encryption and protection** → **Encrypted drives**.

   A list of encrypted drives appears.
3. Select the drive to which the user requested access.
4. Click the **Grant access to the device in offline mode** button.

5. In the window that opens, select the Kaspersky Endpoint Security for Windows plug-in.

6. Follow the instructions provided in the Kaspersky Endpoint Security for Windows Help (see the instructions for OSMP Console at the end of the section).

After that, the user applies the received file to access the encrypted drive and read data stored on the drive.

Transmitting encryption keys between Administration Servers

If the data encryption feature is enabled on a managed device, the encryption key is stored on the Administration Server. The encryption key is used to access encrypted data and to manage the encryption policy.

The encryption key must be transmitted to another Administration Server in the following cases:

- Reconfiguring Network Agent on a managed device to assign the device to another Administration Server. If this device contains encrypted data, the encryption key must be transmitted to the target Administration Server. Otherwise, the data cannot be decrypted.

- Encrypting a removable drive connected to a device D1 that is managed by the Administration Server S1, and then you connect this removable drive to a device D2 managed by the Administration Server S2. To access to the data on the removable drive, the encryption key must be transmitted from the Administration Server S1 to the Administration Server S2.

- Encrypting a file on a device D1 managed by the Administration Server S1, and then you try to access the file on a device D2 managed by the Administration Server S2. To access the file, the encryption key must be transmitted from the Administration Server S1 to the Administration Server S2.

You can transmit encryption keys the following ways:

- Automatically, by enabling the **Use hierarchy of Administration Servers to obtain encryption keys** option in the properties of two Administration Servers between which an encryption key must be transmitted. If this option is disabled for one of the Administration Servers, the automatic transmission of encryption keys is not possible.

  When you enable the **Use hierarchy of Administration Servers to obtain encryption keys** option in an Administration Server properties, the Administration Server sends all of the encryption keys stored in its repository to the primary Administration Server (if any) one level up in the hierarchy.

  When you try to access encrypted data, the Administration Server first searches the encryption key in its own repository. If the **Use hierarchy of Administration Servers to obtain encryption keys** option is enabled and the required encryption key has not been found in the repository, the Administration Server additionally sends a request to the primary Administration Servers (if any) to provide the required encryption key. The request will be sent to all of the primary Administration Servers up to the server on the highest level of the hierarchy.

- Manually from one Administration Server to another by exporting and importing the file containing the encryption keys.

  The export and import of encryption keys are actions that are included in the Encryption key management feature. To perform these actions, configure the access rights to the feature for users of Open Single Management Platform as follows:

  - Grant the **Read** access right to the Encryption key management feature for a user that exports encryption keys from the secondary Administration Server.

  - Grant the **Write** access right to the Encryption key management feature for a user that imports encryption keys to the target Administration Server.

To enable automatic transmission of encryption keys between Administration Servers within the hierarchy:

1. Go to the properties of the required Administration Server:

   a. In the main menu, select a section (except for **Marketplace**, **Console navigation**, and **Settings**), and then at the top of the page, click the **Show hierarchy** button.

   b. In the navigation pane that opens, hover the mouse cursor over the name of the required Administration Server, and then click the settings icon (⚙).

      Alternately, you can click the Administration Server name, and then, after the name is displayed in the main menu, click it in the main menu to open the server properties window.

2. In the properties window that opens, select the **Encryption algorithm** section.

3. Enable the **Use hierarchy of Administration Servers to obtain encryption keys** option.

4. Click **OK** to apply the changes.

The encryption keys will be transmitted to primary Administration Servers (if any) at the next synchronization (the heartbeat). This Administration Server will also provide, upon request, an encryption key from its repository to a secondary Administration Server.

To transmit encryption keys between Administration Servers manually:

1. Go to the properties of the Administration Server from which you want to export encryption keys:

   a. In the main menu, select a section (except for **Marketplace**, **Console navigation**, and **Settings**), and then at the top of the page, click the **Show hierarchy** button.

   b. In the navigation pane that opens, hover the mouse cursor over the name of the required Administration Server, and then click the settings icon (⚙).

      Alternately, you can click the Administration Server name, and then, after the name is displayed in the main menu, click it in the main menu to open the server properties window.

2. In the properties window that opens, select the **Encryption algorithm** section.

3. Click the **Export encryption keys from Administration Server**.

   Make sure that a user that exports encryption keys from the Server is granted the **Read** access right to the Encryption key management feature.

4. In the **Export encryption keys** window:

   - Click the **Browse** button, and then specify where to save the file.

   - Specify a password to protect the file from unauthorized access.

     > Remember the password. A lost password cannot be retrieved. If the password is lost, you have to repeat the export procedure. Therefore, make a note of the password and keep it handy.

5. Transmit the file to another Administration Server, for example, through a shared folder or removable drive.

6. Go to the properties of the Administration Server to which you want to import the encryption keys.

7. In the properties window, select the **Encryption algorithm** section.

8. Click **Import encryption keys to Administration Server**.

   Make sure that a user that imports encryption keys to the Server is granted the **Write** access right to the Encryption key management feature.

9. In the **Import encryption keys** window:

   - Click the **Browse** button, and then select the file containing encryption keys.

   - Specify the password.

10. Click **OK**.

The encryption keys are transmitted to the target Administration Server.

Changing the Administration Server for client devices

You can change the Administration Server to a different one for specific client devices. For this purpose, use the *Change Administration Server* task.

To change the Administration Server that manages client devices to a different Server:

1. Connect to the Administration Server that manages the devices.

2. Create the Administration Server change task.

   The New task wizard starts. Follow the instructions of the wizard. In the **New task** window of the New task wizard, select the **Kaspersky Security Center 16.2** application and the **Change Administration Server** task type. After that, specify the devices for which you want to change the Administration Server:

   - **Assign task to an administration group**

     The task is assigned to devices included in an administration group. You can specify one of the existing groups or create a new one.

     For example, you may want to use this option to run a task of sending a message to users if the message is specific for devices included in a specific administration group.

     If a task is assigned to an administration group, the **Security** tab is not displayed in the task properties window because group tasks are subject to the security settings of the groups to which they apply.

   - **Specify device addresses manually or import addresses from a list**

     You can specify DNS names, IP addresses, and IP subnets of devices to which you want to assign the task.

     You may want to use this option to execute a task for a specific subnet. For example, you may want to install a certain application on devices of accountants or to scan devices in a subnet that is probably infected.

- **Assign task to a device selection**

  The task is assigned to devices included in a device selection. You can specify one of the existing selections.

  For example, you may want to use this option to run a task on devices with a specific operating system version.

3. Run the created task.

   After the task is completed, the client devices for which it was created are put under the management of the Administration Server specified in the task settings.

If the Administration Server supports encryption and data protection and you are creating a *Change Administration Server* task, a warning is displayed. The warning states that if any encrypted data is stored on devices, after the new Server begins managing the devices, users will be able to access only the encrypted data with which they previously worked. In other cases, no access to encrypted data is provided. For detailed descriptions of scenarios in which access to encrypted data is not provided, refer to the Kaspersky Endpoint Security for Windows Help.

Viewing and configuring the actions when devices show inactivity

If client devices within a group are inactive, you can get notifications about it. You can also automatically delete such devices.

To view or configure the actions when the devices in the group show inactivity:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.

2. Go to the properties of the required administration group in one of the following ways:

   - Click the name of the administration group.

   - Click the **Show hierarchy** button at the top of the page. In the navigation pane that opens, hover the mouse cursor over the name of the required group, and then click the settings icon (⚙).

   The administration group properties window opens.

3. In the properties window, go to the **Settings** tab.

4. In the **Inheritance** section, enable or disable the following options:

   - **Inherit from parent group**

     The settings in this section will be inherited from the parent group in which the client device is included. If this option is enabled, the settings under **Device activity on the network** are locked from any changes.

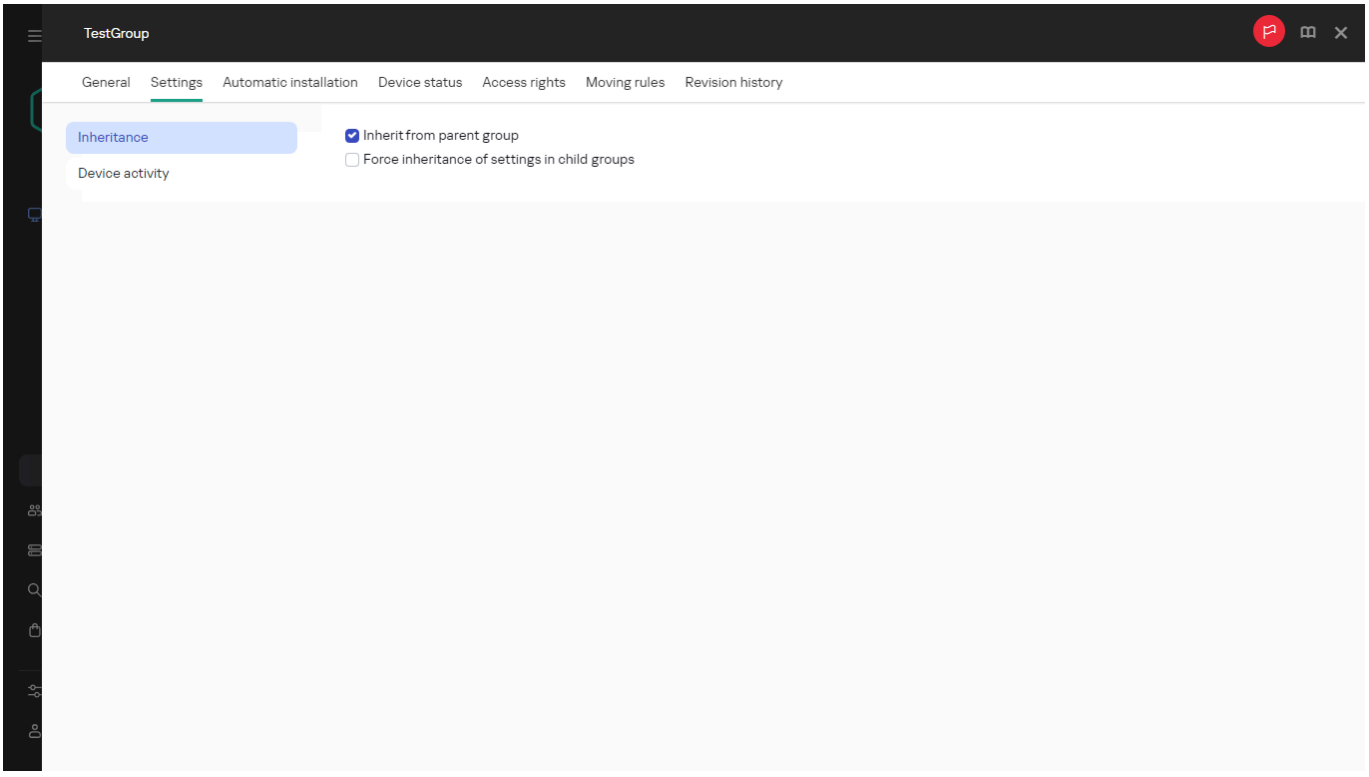
     This option is available only if the administration group has a parent group.

     By default, this option is enabled.

- **Force inheritance of settings in child groups**

  The setting values will be distributed to child groups but in the properties of the child groups these settings are locked.

  By default, this option is disabled.



Administration group properties

5. In the **Device activity** section, enable or disable the following options:

- **Notify the administrator if the device has been inactive for longer than (days)**

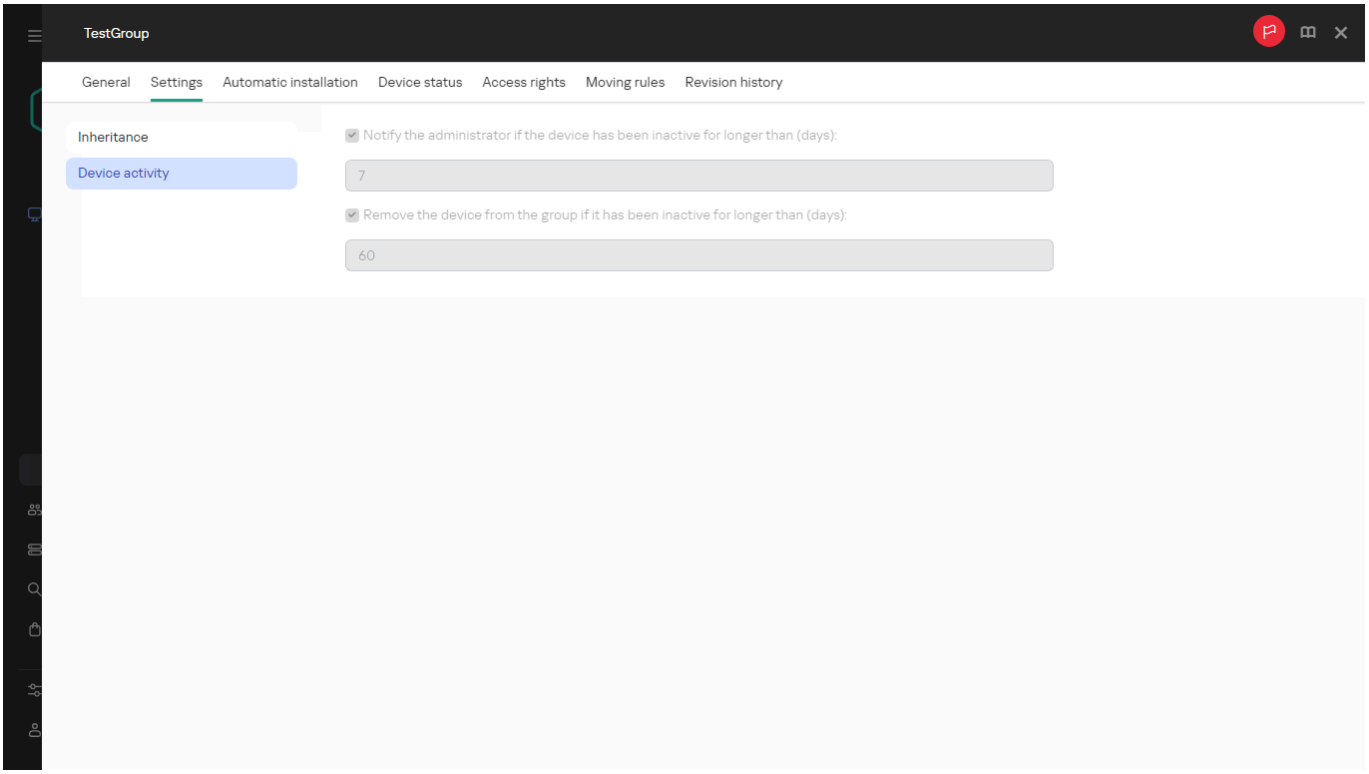
  If this option is enabled, the administrator receives notifications about inactive devices. You can specify the time interval after which the **Device has remained inactive on the network in a long time** event is created. The default time interval is 7 days.

  By default, this option is enabled.

- **Remove the device from the group if it has been inactive for longer than (days)**

  If this option is enabled, you can specify the time interval after which the device is automatically removed from the group. The default time interval is 60 days.

By default, this option is enabled.



Administration group properties

6. Click **Save**.

Your changes are saved and applied.

Remote access to managed devices

Open Single Management Platform allows you to establish a remote connection to managed devices. The administrator can obtain remote access to the managed device through a Network Agent installed on the device. Remote connection to a device through the Network Agent is possible even if the TCP and UDP ports of the client device are closed. Upon establishing a connection with the device, the administrator gains full access to information stored on this device and can manage applications installed on it.

The methods for establishing a remote connection depend on which platform is used on the device with OSMP Console. OSMP Console can be installed on a device other than the Administration Server device, on a Windows- or a Linux-based device.

From OSMP Console installed on a Linux device, you can establish the following connections:

- To a Linux-based managed device by using:
  - Remote desktop connection (RDP)
  - Virtual Network Computing (VNC)
- To a Windows-based managed device by using:
  - Remote desktop connection (RDP)
  - Virtual Network Computing (VNC)

From OSMP Console installed on a Windows device, you can establish the following connections:

- To a Linux-based managed device by using:
  - Remote desktop connection (RDP)
  - Virtual Network Computing (VNC)
- To a Windows-based managed device by using:
  - Remote desktop connection (RDP)
  - Windows Desktop Sharing (WDS)

To establish a remote connection to a device, you must have the following:

- klsctunnel utility

  The Kaspersky utility is used for tunneling the connection between a managed device and Administration Server.

  Open Single Management Platform allows tunneling TCP connections from OSMP Console via the Administration Server and then via Network Agent to a specified port on a managed device. Tunneling is designed for connecting a client application on a device with OSMP Console installed to a TCP port on a managed device—if no direct connection is possible between OSMP Console and the target device.

  Connection tunneling between a remote client device and Administration Server is required if the port used for connection to Administration Server is not available on the device. The port on the device may be unavailable in the following cases:

  - The remote device is connected to a local network that uses the NAT mechanism.
  - The remote device is part of the local network of Administration Server, but its port is closed by a firewall.

  The utility can be downloaded when a remote connection is established through OSMP Console or manually. The klsctunnel utility is installed on the Administration Server device.

  The klsctunnel utility is used for all types of the remote connection (RDP, VNC, WDS).

- RDP or VNC client

  An RDP or VNC client is installed on the Administration Server device. This client allows you to obtain remote access to the managed device by using the address and port for the local connection provided by the klsctunnel utility.

  An RDP connection to the current remote desktop session of the user is established without the user's knowledge. Once the administrator connects to the session, the device user is disconnected from the session without notification.

- Windows Desktop Sharing (only for Windows-based managed devices)

  This feature allows you to connect to an existing session on a client device without disconnecting the user in this session. In this case, the administrator and the session user on the device share access to the desktop.

  When connecting to an existing session of the remote desktop, the session user on the device receives a connection request from the administrator. No information about remote activity on the device and its results will be saved in reports created by Open Single Management Platform.

  The administrator can configure an audit of user activity on a remote client device. During the audit, the application saves information about files on the client device that have been opened and/or modified by the administrator.

Remote access from a Linux-based device with OSMP Console to a Linux-based managed device

Prerequisites

Before you start, make sure that you have done the following:

- On the managed device running Linux, Network Agent version 15.3 or later is installed.
- On the managed device running Linux, the RDP server is installed (if you connect to the managed device by using Remote Desktop Connection).
- On the managed device running Linux, the VNC server is installed (if you connect to the managed device by using Virtual Network Computing).
- The administrator has the **Start RDP sessions** right to establish RDP connections (if you connect to the managed device by using Remote Desktop Connection).
- The administrator has the **Initiate tunneling** right to establish the tunnel connection for VNC (if you connect to the managed device by using Virtual Network Computing).

To connect to a Linux-based managed device by using RDP or VNC through OSMP Console opened on a Linux-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.
2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

   The **Connect to Remote Desktop** window opens.

   If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.
3. Download the klsctunnel utility by clicking the **Download** button, and then run it.

   When downloading the utility, take into account the following:

   - To get an up-to-date version of the klsctunnel utility, make sure that Administration Server has access to Kaspersky update servers.

     You can run the *Download updates to the Administration Server repository* task and select Kaspersky update servers as a source of updates. If the task finishes successfully, Administration Server has access to Kaspersky update servers.
   - If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually, and then place it on the Administration Server device in the /var/opt/kaspersky/klnagent_srv directory.
4. In the **Connect to Remote Desktop** window, specify the connection port. By default, port 3389 is used.

5. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.

6. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   The klsctunnel utility displays the address and port for the local connection of a remote access client.

   The utility allows the administrator to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

7. Run a remote access client (RDP or VNC) and connect it to the managed device by using the address and port provided by the klsctunnel utility.

A connection to the managed device is established, and the desktop is available in the remote access client window.

Remote access from a Linux-based device with OSMP Console to a Windows-based managed device

Remote desktop connection (RDP)

Prerequisites

Before you start, make sure that you have done the following:

- Remote connection is allowed in the operating system settings of the managed device.
- On the managed device running Windows, Network Agent version 13.0 or later is installed.
- The administrator has the **Start RDP sessions** right to establish RDP connections.

To connect to a Windows-based managed device by using RDP through OSMP Console opened on a Linux-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.

2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

   The **Connect to Remote Desktop** window opens.

   If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.

3. In the **Connect to Remote Desktop** window, select the **RDP** connection type.

4. If the remote connection is not allowed in the operating system settings of the managed device, allow remote connection centrally by clicking the **Change settings** button.

   If the settings are applied correctly, a notification is displayed. Also, on the managed device, in the **Settings** → **System** → **Remote Desktop** section, the **Enable Remote Desktop** option is enabled.

   If the remote connection is allowed in the operating system settings of the managed device, the **Change settings** button is not displayed.

5. Download the klsctunnel utility by clicking the **Download** button, and then run it.

   If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually.

6. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.

7. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   The klsctunnel utility displays the address and port of the connection to the remote device.

   The utility allows the administrator to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

8. Run an RDP client and connect it to the managed device by using the address and port provided by the klsctunnel utility.

A connection to the managed device is established, and the desktop is available in the RDP client window.

Connection to the current remote desktop session of the user is established without the user's knowledge. Once the administrator connects to the session, the device user is disconnected from the session without notification.

Virtual Network Computing system (VNC)

Prerequisites

Before you start, make sure that you have done the following:

- On the managed device running Windows, Network Agent version 13.0 or later is installed.
- On the managed device running Windows, the VNC server is installed.
- The administrator has the **Initiate tunneling** right to establish the tunnel connection for VNC.

To connect to a Windows-based managed device by using VNC through OSMP Console opened on a Linux-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.

2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

   The **Connect to Remote Desktop** window opens.

   If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.

3. In the **Connect to Remote Desktop** window, select the connection type **VNC**.

4. Download the klsctunnel utility by clicking the **Download** button, and then run it.

When downloading the utility, take into account the following:

- To get an up-to-date version of the klsctunnel utility, make sure that Administration Server has access to Kaspersky update servers.

  You can run the *Download updates to the Administration Server repository* task and select Kaspersky update servers as a source of updates. If the task finishes successfully, Administration Server has access to Kaspersky update servers.

- If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually, and then place it on the Administration Server device in the /var/opt/kaspersky/klnagent directory.

5. In the **Connect to Remote Desktop** window, specify the VNC connection port. By default, port 5900 is used.

6. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.

7. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   The klsctunnel utility displays the address and port for the local connection of the VNC client.

   The utility allows the remote user to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

8. Run a VNC client and connect it to the managed device by using the address and port provided by the klsctunnel utility.

Connection to the managed device is established, and the desktop is available in the VNC client window.

Remote access from a Windows-based device with OSMP Console to a Linux-based managed device

Remote desktop connection (RDP)

Prerequisites

Before you start, make sure that you have done the following:

- On the managed device running Linux, the RDP server is installed.
- On the managed device running Linux, Network Agent version 15.3 or later is installed.
- The administrator has the **Start RDP sessions** right to establish RDP connections.

To connect to a Linux-based managed device by using RDP through OSMP Console opened on a Windows-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.
2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

The **Connect to Remote Desktop** window opens.

If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.

3. In the **Connect to Remote Desktop** window, select the **RDP** connection type.

4. Download the klsctunnel utility by clicking the **Download** button, and then run it.

   If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually.

5. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.

6. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   To do this, select the **Use proxy server** check box, and then specify the connection settings.

7. Click the **Open port** button.

   The Remote Desktop Connection login window opens.

8. Specify the credentials of the account under which you are currently logged in to OSMP Console, and then connect to the managed device.

   The klsctunnel utility displays the address and port of the connection to the remote device.

   The utility allows the administrator to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

When connection to the device is established, the desktop is available in the Remote Desktop Connection window of Microsoft Windows.

Connection to the current remote desktop session of the user is established without the user's knowledge. Once the administrator connects to the session, the device user is disconnected from the session without notification.

Virtual Network Computing system (VNC)

Prerequisites

Before you start, make sure that you have done the following:

- On the managed device running Linux, Network Agent version 15.3 or later is installed.
- On the managed device running Linux, the VNC server is installed.
- The administrator has the **Initiate tunneling** right to establish the tunnel connection for VNC.

To connect to a Linux-based managed device by using VNC through OSMP Console opened on a Windows-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.

2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

The **Connect to Remote Desktop** window opens.

If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.

3. In the **Connect to Remote Desktop** window, select the connection type **VNC**.

4. Download the klsctunnel utility by clicking the **Download** button, and then run it.

   When downloading the utility, take into account the following:

   - To get an up-to-date version of the klsctunnel utility, make sure that Administration Server has access to Kaspersky update servers.

     You can run the *Download updates to the Administration Server repository* task and select Kaspersky update servers as a source of updates. If the task finishes successfully, Administration Server has access to Kaspersky update servers.

   - If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually, and then place it on the Administration Server device in the /var/opt/kaspersky/klnagent directory.

5. In the **Connect to Remote Desktop** window, specify the VNC connection port. By default, port 5900 is used.

6. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.

7. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   To do this, select the **Use proxy server** check box, and then specify the connection settings.

8. Click the **Open port** button.

   The klsctunnel utility displays the address and port for the local connection of the VNC client.

   The utility allows the remote user to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

9. Run a VNC client and connect it to the managed device by using the address and port provided by the klsctunnel utility.

Connection to the managed device is established, and the desktop is available in the VNC client window.

Remote access from a Windows-based device with OSMP Console to a Windows-based managed device

Remote desktop connection (RDP)

Prerequisites

Before you start, make sure that you have done the following:

- On the managed device running Windows, Network Agent version 13.0 or later is installed.
- The administrator has the **Start RDP sessions** right to establish RDP connections.
- A remote connection is allowed in the operating system settings of the managed device.

To connect to a Windows-based managed device by using RDP through OSMP Console opened on a Windows-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.
2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

   The **Connect to Remote Desktop** window opens.

   If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.
3. In the **Connect to Remote Desktop** window, select the **Remote Desktop Connection** connection type.
4. If a remote connection is not allowed in the operating system settings of the managed device, allow the remote connection centrally by clicking the **Change settings** button.

   If the settings are applied correctly, a notification is displayed. Also, on the managed device, in the **Settings** → **System** → **Remote Desktop** section, the **Enable Remote Desktop** option is enabled.

   If the remote connection is allowed in the operating system settings of the managed device, the **Change settings** button is not displayed.
5. Download the klsctunnel utility by clicking the **Download** button, and then run it.

   If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually.
6. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.
7. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   To do this, select the **Use proxy server** check box, and then specify the connection settings.
8. Click the **Open port** button.

   The Remote Desktop Connection login window opens.
9. Specify the credentials of the account under which you are currently logged in to OSMP Console, and then connect to the managed device.

   The klsctunnel utility displays the address and port of the connection to the remote device.

   The utility allows the administrator to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

When connection to the device is established, the desktop is available in the Remote Desktop Connection window of Microsoft Windows.

Connection to the current remote desktop session of the user is established without the user's knowledge. Once the administrator connects to the session, the device user is disconnected from the session without notification.

Windows Desktop Sharing (WDS)

Prerequisites

Before you start, make sure that you have done the following:

- On the managed device running Windows, Network Agent version 13.0 or later is installed.
- The administrator has the **Connect to existing RDP sessions** right to establish the WDS connection.
- The license for Vulnerability and patch management is available.
- Microsoft Windows Vista or later is installed on the administrator's workstation. The type of operating system of the device hosting Administration Server imposes no restrictions on connection through Windows Desktop Sharing.

  Check whether the Windows Desktop Sharing feature is included in your Windows edition, and make sure that the CLSID\{32BE5ED2-5C86-480F-A914-0FF8885A1B3F} key exists in the Windows Registry.
- Microsoft Windows Vista or later is installed on the client device.

To connect to a Windows-based managed device by using VNC through OSMP Console opened on a Windows-based device:

1. In the main menu, go to the **Asset management** → **Assets (Devices)** section or open a device selection.
2. Select the check box next to the managed device to which you want to connect remotely, and then click the **Connect to Remote Desktop** button.

   The **Connect to Remote Desktop** window opens.

   If you select multiple devices, a mobile device, or a device running macOS, the **Connect to Remote Desktop** button will be disabled.
3. In the **Connect to Remote Desktop** window, select the connection type **Windows Desktop Sharing**.

   If the Vulnerability and patch management license is not available, an error message is displayed.
4. Download the klsctunnel utility by clicking the **Download** button, and then run it.

   When downloading the utility, take into account the following:

   - To get an up-to-date version of the klsctunnel utility, make sure that Administration Server has access to Kaspersky update servers.

     You can run the *Download updates to the Administration Server repository* task and select Kaspersky update servers as a source of updates. If the task finishes successfully, Administration Server has access to Kaspersky update servers.
   - If the utility file is unavailable for download, an error message is displayed. In this case, download the utility manually, and then place it on the Administration Server device in the /var/opt/kaspersky/klnagent directory.

5. In the list of available user sessions active on the selected device, select the session to which you want to connect.

   The remote user must allow the connection. If the remote user refuses to connect or if the remote user does not allow the connection during the timeout period, an error message is displayed.

6. Generate a text blob with encoded connection parameters by clicking the **Generate blob** button, and then copy and paste the text into the corresponding field in the klsctunnel utility.

   A blob contains the settings required to establish a connection between Administration Server and the managed device. A blob is valid for three minutes. If it has expired, generate a new blob.

7. In the klsctunnel utility, if you use a proxy server, specify the proxy server connection settings.

   To do this, select the **Use proxy server** check box, and then specify the connection settings.

8. Click the **Open port** button.

   The klsctunnel utility displays the address and port for the local connection of the VNC client.

   The utility allows the remote user to close the tunnel connection. If the tunnel connection is closed, the current connection to the remote desktop is terminated.

Desktop sharing starts in a new window. If you want to interact with the device, click the menu icon ( ) in the upper-left corner of the window, and then select **Interactive mode**.

Managing assets

Assets represent the computers of the organization. You can add assets to OSMP; in that case, OSMP automatically adds asset IDs when enriching events, and when you analyze events, you can get additional information about computers in the organization.

You can add assets to OSMP in the following ways:

- Import assets:
  - From the MaxPatrol report.
  - On a schedule from **Kaspersky Security Center** and KICS for Networks.

    All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

    KUMA imports the following device information received from Kaspersky Security Center Network Agents:

    - Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
    - Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Click **Import**, and then select one of the following actions:
   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.
   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.
3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

   If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.
4. Click the **Import** button.

The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

By default, assets are imported every 12 hours, this frequency can be configured. On-demand import of assets is also possible; such on-demand import does not affect the scheduled import time. From the Kaspersky Security Center database, OSMP imports information about devices with installed Kaspersky Security Center Network Agent that has connected to Kaspersky Security Center, that is, has a non-empty 'Connection time' field in the SQL database. OSMP imports the following information about the computer: name, address, time of connection to Kaspersky Security Center, information about hardware and software, including the operating system, as well as vulnerabilities, that is, information received from Kaspersky Security Center Network Agents.

- Create assets manually through the web interface or via the API.

  You can add assets manually. In this case, you must manually specify the following information: address, FQDN, name and version of the operating system, hardware information. Information about the vulnerabilities of assets cannot be added through the web interface. You can provide information about vulnerabilities if you add assets using the API.

You can manage OSMP assets: view information about assets, search for assets, add, edit or delete assets, and export asset data to a CSV file.

Asset categories

You can categorize the assets and then use the categories in filter conditions or correlation rules. For example, you can create alerts of a higher severity level for assets from a higher-severity category. By default, all assets fall into the **Uncategorized assets** category. A device can be added to multiple categories.

By default, OSMP assigns the following severity levels to asset categories: Low, Medium, High, Critical. You can create custom categories, categories can be nested.

Categories can be populated in the following ways:

- **Manually**
- **Active**: dynamic if the asset **meets the specified conditions**. For example, the moment the asset is upgraded to a specified OS version or placed in a specified subnet, the asset is moved to the specified category.

1. In the **Repeat categorization every** drop-down list, specify how often assets will be linked to a category. You can select values ranging from once per hour to once per 24 hours.

   You can forcibly start categorization by selecting **Start categorization** in the category context menu.

2. In the **Conditions** settings block, specify the filter for matching assets to attach to an asset category.

   You can add conditions by clicking the **Add condition** buttons. Groups of conditions can be added by using the **Add group** buttons. Group operators can be switched between **AND**, **OR**, and **NOT** values.

   Categorization filter operands and operators

Operand	Operators	Comment
Build number	=, ilike	
OS	=, ilike	The **ilike** operator makes the search case-insensitive.
IP address	inSubnet, inRange	The IP address is indicated in CIDR notation (for example: 192.168.0.0/24). When the inRange operator is selected, you can indicate only addresses from private ranges of IP addresses (for example: 10.0.0.0–10.255.255.255). Both addresses must be in the same range.
FQDN	=, ilike	The **ilike** operator makes the search case-insensitive.
CVE	=, in	The **in** operator lets you specify an array of CVE (Common Vulnerabilities and Exposures) IDs.
CVSS	>, >=, =, <=,<	Severity level of CVE vulnerabilities on the asset. The CVSS parameter takes values from 0 to 10. Not applicable to vulnerabilities from Open Single Management Platform.

CVE count	>, >=, =, <=, <	The number of unique vulnerabilities with the CVE attribute for the asset. Vulnerabilities without CVEs do not count towards this figure.
		For categorization by the number of CVEs of a certain severity level, you can use a combined condition. For example:
		`CVE count >= 1`
		`AND`
		`CVSS >= 6.5`

Software	=, ilike	Categorization by software installed on the asset. The **ilike** operator makes the search case-insensitive.
Software version	=, ilike, in	Categorization by version (build) number of the software installed on the asset. The **ilike** operator makes the search case-insensitive.
CII	in	More than one value can be selected.
KSC group	=, ilike	Categorization by the name of the Open Single Management Platform administration group in which the asset is placed.
Anti-virus databases last updated	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is current at the time when categorization is started.

Last update of the information	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
Protection last updated	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
System last started	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
KSC extended status	in	Extended status of the device. More than one value can be selected.
Real-time protection status	=	Status of Kaspersky applications installed on the managed device.
Encryption status	=	
Spam protection status	=	
Anti-virus protection status of mail servers	=	
Data Leakage Prevention status	=	
KSC extended status ID	=	
Endpoint Sensor status	=	
Last visible	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
Score ML	>,>=,=,<=,<	Categorization by asset score assigned by AI services.
Status	=, in	Categorization by predefined asset statuses assigned by AI services.
Custom asset field	=, ilike	Categorization by values of custom asset fields.

Using time values

Some conditions, for example, **Anti-virus databases last updated** or **System last started**, use date and time as the operand value. For these conditions, you can use an exact date and time or a relative period.

To specify a date and time value:

1. Select an operand, an operator and click the date field.

2. Do one of the following:

   - Select the exact date in the calendar.

     By default, the current time is automatically added to the selected date, with millisecond precision. Changing the date in the calendar does not change the specified time. The date and time are displayed in the time zone of the browser. If necessary, you can edit the date and time in the field.

   - In the **Relative period** list, select a relative period.

     The period is calculated relative to the start time of the current categorization and takes into account asset information that is up-to-date at that moment. For example, for the condition **Anti-virus databases last updated**, you can select **1 hour** and the **>=** operator to periodically link to the category those assets for which the anti-virus databases have not been updated for more than 1 hour before the start of categorization.

   - In the date and time field, enter a value manually.

     You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a relative period as a formula. You can also combine these methods if necessary.

     If you do not specify milliseconds when entering the exact date, 000 is substituted automatically.

     In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+**, **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second).

     For example, for the **Information last updated** condition, you can specify the value **now-2d** with the operator **>=** operator and the value **now-1d** with the **>=** operator to regularly link assets to the category if those assets had information updated during the day before the categorization was started; alternatively, you can specify the value **now/w** with the **<=** operator to regularly link assets to the category if those assets had information updated between the beginning of the first day of the current week (00:00:00:000 UTC) and **now**.

     KUMA stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the period, the category will cover assets from 03:00:00.000 until now, not from 00:00:00.000 until now.

     If you want to take your time zone into account when selecting a relative period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the date and time field by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want the categorization to cover the **Yesterday** period, you need to change the value to **now-1d/d-3h**. If you want the categorization to cover the **Today** period, change the value to **now/d-3h**.

3. Use the **Test conditions** button to make sure that the specified filter is correct. When you click the

button, you should see the **Assets for given conditions** window containing a list of assets that satisfy the search conditions.

- **Reactive**—When a correlation rule is triggered, the asset is moved to the specified group.

In OSMP, assets are categorized by tenant and by category. Assets are arranged in a tree structure, where the tenants are located at the root, and the asset categories branch from them. You can view the tree of tenants and categories in the **Assets** → **All assets** section. When a tree node is selected, the assets assigned to it are displayed in the right part of the window. Assets from the subcategories of the selected category are displayed if you specify that you want to display assets recursively. You can select the check boxes next to the tenants whose assets you want to view.

To open the context menu of a category, hover the mouse cursor over the category and click the ellipsis icon that is displayed to the right of the category name. The following actions are available in the context menu:

Category context menu items

Action	Description
Show assets	Display assets of the selected category in the right part of the window.
Show assets recursively	View assets from subcategories of the selected category. If you want to exit recursive viewing mode, select another category to view.
Show info	View information about the selected category in the **Category information** details area displayed in the right part of the web interface window.
Start categorization	Start automatic binding of assets to the selected category. This option is available for categories that have active categorization.
Add subcategory	Add a subcategory to the selected category.
Edit category	Edit the selected category.
Delete category	Delete the selected category. You can only delete categories that have no assets or subcategories. Otherwise the **Delete category** option is inactive.
Pin as tab	Display the selected category on a separate tab. You can undo this action by selecting **Unpin as tab** in the context menu of the relevant category.

Adding an asset category

To add an asset category:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.

2. Open the category creation window:

   - Click the **Add category** button.

   - If you want to create a subcategory, select **Add subcategory** in the context menu of the parent category.

   The **Add category** details area appears in the right-hand part of the console window.

3. Add information about the category:

- In the **Name** field, enter the name of the category. The name must contain 1 to 128 Unicode characters.
- In the **Parent** field, indicate the position of the category within the categories tree hierarchy:

  a. Click the button.

  This opens the **Select categories** window showing the categories tree. If you are creating a new category and not a subcategory, the window may show multiple asset category trees, one for each tenant that you can access. Your tenant selection in this window cannot be undone.

  b. Select the parent category for the category you are creating.

  c. Click **Save**.

  Selected category appears in **Parent** fields.

- The **Tenant** field displays the tenant whose structure contains your selected parent category. The tenant category cannot be changed.
- Assign a severity to the category in the **Priority** drop-down list.
- If necessary, in the **Description** field, you can add a note consisting of up to 256 Unicode characters.

4. In the **Categorization kind** drop-down list, select how the category will be populated with assets. Depending on your selection, you may need to specify additional settings:

- **Manually**—assets can only be manually linked to a category.
- **Active**—assets will be assigned to a category at regular intervals if they satisfy the defined filter.

  Active category of assets

  1. In the **Repeat categorization every** drop-down list, specify how often assets will be linked to a category. You can select values ranging from once per hour to once per 24 hours.

     You can forcibly start categorization by selecting **Start categorization** in the category context menu.

  2. In the **Conditions** settings block, specify the filter for matching assets to attach to an asset category.

     You can add conditions by clicking the **Add condition** buttons. Groups of conditions can be added by using the **Add group** buttons. Group operators can be switched between **AND**, **OR**, and **NOT** values.

     Categorization filter operands and operators

Operand	Operators	Comment
Build number	=, ilike	
OS	=, ilike	The **ilike** operator makes the search case-insensitive.
IP address	inSubnet, inRange	The IP address is indicated in CIDR notation (for example: 192.168.0.0/24). When the inRange operator is selected, you can indicate only addresses from private ranges of IP addresses (for example: 10.0.0.0–10.255.255.255). Both addresses must be in the same range.
FQDN	=, ilike	The **ilike** operator makes the search case-insensitive.

CVE	=, in	The **in** operator lets you specify an array of CVE (Common Vulnerabilities and Exposures) IDs.
CVSS	>, >=, =, <=,<	Severity level of CVE vulnerabilities on the asset. The CVSS parameter takes values from 0 to 10. Not applicable to vulnerabilities from Open Single Management Platform.

CVE count	>, >=, =, <=, <	The number of unique vulnerabilities with the CVE attribute for the asset. Vulnerabilities without CVEs do not count towards this figure.
		For categorization by the number of CVEs of a certain severity level, you can use a combined condition. For example:
		`CVE count >= 1`
		`AND`
		`CVSS >= 6.5`

Software	=, ilike	Categorization by software installed on the asset. The **ilike** operator makes the search case-insensitive.
Software version	=, ilike, in	Categorization by version (build) number of the software installed on the asset. The **ilike** operator makes the search case-insensitive.
CII	in	More than one value can be selected.
KSC group	=, ilike	Categorization by the name of the Open Single Management Platform administration group in which the asset is placed.
Anti-virus databases last updated	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is current at the time when categorization is started.
Last update of the information	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
Protection last updated	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
System last started	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.

KSC extended status	in	Extended status of the device. More than one value can be selected.
Real-time protection status	=	Status of Kaspersky applications installed on the managed device.
Encryption status	=	
Spam protection status	=	
Anti-virus protection status of mail servers	=	
Data Leakage Prevention status	=	
KSC extended status ID	=	
Endpoint Sensor status	=	
Last visible	>=,<=	For categorization The time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below. A relative period for repeated categorization takes into account asset information that is up-to-date at the time when categorization is started.
Score ML	>,>=,=,<=,<	Categorization by asset score assigned by AI services.
Status	=, in	Categorization by predefined asset statuses assigned by AI services.
Custom asset field	=, ilike	Categorization by values of custom asset fields.

Using time values

Some conditions, for example, **Anti-virus databases last updated** or **System last started**, use date and time as the operand value. For these conditions, you can use an exact date and time or a relative period.

To specify a date and time value:

1. Select an operand, an operator and click the date field.

2. Do one of the following:

   - Select the exact date in the calendar.

     By default, the current time is automatically added to the selected date, with millisecond precision. Changing the date in the calendar does not change the specified time. The date and time are displayed in the time zone of the browser. If necessary, you can edit the date and time in the field.

   - In the **Relative period** list, select a relative period.

     The period is calculated relative to the start time of the current categorization and takes into account asset information that is up-to-date at that moment. For example, for the condition **Anti-virus databases last updated**, you can select **1 hour** and the **>=** operator to periodically link to the category those assets for which the anti-virus databases have not been updated for more than 1 hour before the start of categorization.

- In the date and time field, enter a value manually.

  You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a relative period as a formula. You can also combine these methods if necessary.

  If you do not specify milliseconds when entering the exact date, 000 is substituted automatically.

  In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+**, **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second).

  For example, for the **Information last updated** condition, you can specify the value **now-2d** with the operator **>=** operator and the value **now-1d** with the **>=** operator to regularly link assets to the category if those assets had information updated during the day before the categorization was started; alternatively, you can specify the value **now/w** with the **<=** operator to regularly link assets to the category if those assets had information updated between the beginning of the first day of the current week (00:00:00:000 UTC) and **now**.

  KUMA stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the period, the category will cover assets from 03:00:00.000 until now, not from 00:00:00.000 until now.

  If you want to take your time zone into account when selecting a relative period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the date and time field by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want the categorization to cover the **Yesterday** period, you need to change the value to **now-1d/d-3h**. If you want the categorization to cover the **Today** period, change the value to **now/d-3h**.

3. Use the **Test conditions** button to make sure that the specified filter is correct. When you click the button, you should see the **Assets for given conditions** window containing a list of assets that satisfy the search conditions.

- **Reactive**—the category will be filled with assets by using correlation rules.

5. Click **Save**.

The new category will be added to the asset categories tree.

Configuring the table of assets

In OSMP, you can configure the contents and order of columns displayed in the assets table. These settings are stored locally on your machine.

To configure the settings for displaying the assets table:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.

2. Click the **Settings** icon (⚙) in the upper-right corner of the assets table.

3. In the drop-down list, select the check boxes next to the parameters that you want to view in the table:

   - **Name**
   - **FQDN**
   - **IP address**
   - **Categories**
   - **MAC address**
   - **Asset source**
   - **OS**
   - **Created on**
   - **Updated on**
   - **Last seen at**
   - **Tenant**
   - **CII category**
   - **Archived**
   - **Weight**
   - **KSC group**
   - **Telemetry received on**
   - **Telemetry status**
   - **EDR status**
   - **Last connection to KSC**
   - **Last connection to EDR**
   - **EDR license**
   - **EDR agent version**
   - **EDR isolation**
   - **AI score**

   When you select a check box, the assets table is updated and a new column is added. When a check box is cleared, the column disappears. The table can be sorted based on multiple columns.

4. If you need to change the order of columns, click the left mouse button on the column name and drag it to the desired location in the table.

The assets table display settings are configured.

Searching assets

OSMP has two asset search modes. You can switch between the search modes using the buttons in the upper left part of the window:

- – simple search by the following asset settings: **Name**, **FQDN**, **IP address**, **MAC address**, and **Owner**.
- – advanced search for assets using filters by conditions and condition groups.

You can select the check boxes next to the found assets to export their details to a CSV file.

Simple search

To find an asset using simple search:

1. In the **Asset Management** section, click the button.

   The **Search** field is displayed at the top of the window.

2. Enter your search query in the **Search** field and press **ENTER** or click the icon.

The table displays the assets with the **Name**, **FQDN**, **IP address**, **MAC address**, and **Owner** settings matching the search criteria.

Advanced search

To find an asset using advanced search:

1. In the **Asset Management** section, click the button.

   The asset filtering settings are displayed in the upper part of the window.

2. Specify the asset filtering settings and click the **Search** button.

   For details on asset filtering settings, see the table below.

The table displays the assets that meet the search criteria.

An advanced asset search is performed using the filtering conditions that can be specified in the upper part of the window:

- You can use the **Add condition** button to add a string containing fields for identifying the condition.
- You can use the **Add group** button to add a group of filters. Group operators can be switched between **AND**, **OR**, and **NOT**.
- Conditions and condition groups can be dragged with the mouse.
- Conditions, groups, and filters can be deleted by using the button.
- You can collapse the filtering options by clicking the **Collapse** button. In this case, the resulting search expression is displayed. Clicking it displays the search criteria in full again.

- The filtering options can be reset by clicking the **Clear** button.

- The condition operators and available values of the right operand depend on the selected left operand:

Left operand	Available operators	Right operand
Build number	=, ilike	An arbitrary value.
OS	=, ilike	An arbitrary value.
IP address	inSubnet, inRange	An arbitrary value or a range of values. The filtering condition for the inSubnet operator is met if the IP address in the left operand is included in the subnet that is specified in the right operand. For example, the subnet for the IP address 10.80.16.206 should be specified in the right operand using slash notation as follows: `10.80.16.206/25`.
FQDN	=, ilike	An arbitrary value.
CVE	=, in	An arbitrary value. The **in** operator lets you specify an array of CVE (Common Vulnerabilities and Exposures) IDs, for example, `CVE-2017-0143`.
CVSS	>, >=, =, <=, <	A number from 0 to 10 (possible severity levels of the asset's CVE vulnerability). Not applicable for vulnerabilities from the Open Single Management Platform.

CVE count	>, >=, =, <=, <	Number. The number of unique vulnerabilities with the CVE attribute for the asset. Vulnerabilities without CVEs do not count towards this figure.
		If `0` is specified as the right operand and the `=` or `<=` operator is selected, the search results include assets that do not have CVE vulnerabilities.
		For searching by the number of CVEs of a certain severity level, you can use a combined condition. For example:
		`CVE count >= 1`
		`CVSS >= 6.5`

Software	=, ilike	An arbitrary value.
Software version	=, ilike, in	An arbitrary value. Version (build) number of the software installed on the asset.
Asset source	in	• Open Single Management Platform. • KICS/KATA. • Created manually
CII	in	• Information resource is not a CII object. • CII object without a significance category. • CII object of the third category of significance. • CII object of the second category of significance. • CII object of the first category of significance.
RAM (bytes)	=, >, >=, <, <=	Number.
Number of disks	=, >, >=, <, <=	Number.
Number of network cards	=, >, >=, <, <=	Number.
Disk free bytes	=, >, >=, <, <=	Number.
KSC group	=, ilike	An arbitrary value. The name of the Open Single Management Platform administration group to which the asset belongs.
Anti-virus databases last updated	>=, <=	For searching, the time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, check the *Using time values* subsection below.
Last update of the information	>=, <=	For searching, the time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below.

Protection last updated	>=, <=	For searching, the time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, check the *Using time values* subsection below.

System last started	>=, <=	For searching, the time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, see the *Using time values* subsection below.
KSC extended status	in	• Host with Network Agent installed is online, but Network Agent is inactive. • Anti-virus application is installed, but real-time protection is not running. • Anti-virus application is installed but not running. • Number of viruses detected is too high. • Anti-virus application is installed but the real-time protection status differs from the one set by the security administrator. • Anti-virus application is not installed. • Full scan for viruses performed too long ago. • Anti-virus bases were updated too long ago. • Network Agent has been inactive too long. • Old license. • Number of uncured objects is too high. • Reboot is required. • One or more incompatible applications are installed on the host. • Host has one or more vulnerabilities. • Last search for operating system updates was performed too long ago on the host. • The host does not have the proper encryption status. • Mobile device settings do not meet the requirements of the security policy. • There are unhandled incidents. • Host status was suggested by the managed product (HSDP). • Host is out of disk space, either synchronization errors occur, or disk space is running out.
Real-time protection status	=	• Unknown. • Stopped. • Suspended. • Starting. • Running (if anti-virus application does not support categories of state Running). • Running with maximum protection. • Running for maximum speed. • Running with recommended settings. • Running with custom settings. • Error.
Encryption status	=	• Encryption rules are not configured on the host. • Encryption is in progress. • Encryption was canceled by the user. • Encryption error occurred. • All host encryption rules are met. • Encryption is in progress, the host must be restarted. • Encrypted files without specified encryption rules are detected on the host.
Spam protection status	=	• Unknown. • Stopped. • Suspended. • Starting. • Running. • Error. • Not installed. • No license.
Anti-virus protection status of mail servers	=	• Unknown. • Stopped. • Suspended. • Starting. • Running. • Error. • Not installed. • No license.
Data Leakage Prevention status	=	• Unknown. • Stopped. • Suspended. • Starting. • Running. • Error. • Not installed. • No license.
KSC extended status ID	=	• OK • Critical.

		• Warning.
Endpoint Sensor status	=	• Unknown. • Stopped. • Suspended. • Starting. • Running. • Error. • Not installed. • No license.
Last visible	>=, <=	For searching, the time is specified as UTC time, and then converted in the KUMA interface to the local time zone set in the browser. You can specify the date and time for this operand in one of the following ways: • Select the exact date in the calendar. • Select a period relative to the present time in the **Relative period** list. • Enter a value manually: an exact date and time or a relative period, or a combination of both. For details, check the *Using time values* subsection below.
Score ML	=, >, >=, <, <=	Number. Asset score assigned by AI services.
Status	=, in	Asset status assigned by AI services: • Low. • Medium. • High. • Critical.
Custom asset field	=, ilike	An arbitrary value. Search custom fields of assets.

Using time values

Some conditions, for example, **Anti-virus databases last updated** or **System last started**, use date and time as the operand value. For these conditions, you can use an exact date and time or a relative period.

To specify a date and time value:

1. Select an operand, an operator and click the date field.

2. Do one of the following:

   - Select the exact date in the calendar.

     By default, the current time is automatically added to the selected date, with millisecond precision. Changing the date in the calendar does not change the specified time. The date and time are displayed in the time zone of the browser. If necessary, you can edit the date and time in the field.

   - In the **Relative period** list, select a relative period.

     The period is calculated relative to the start time of the current search and takes into account asset information that is up-to-date at that moment. For example, for the condition **Anti-virus databases last updated**, you can select **1 hour** and the **>=** operator to find those assets for which the anti-virus databases have not been updated for more than 1 hour.

   - In the date and time field, enter a value manually.

     You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a relative period as a formula. You can also combine these methods if necessary.

     If you do not specify milliseconds when entering the exact date, 000 is substituted automatically.

     In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: **+**, **-**, **/** (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second).

For example, for the **Information last updated** condition, you can specify **now-2d** with the **>=** operator and **now-1d** with the **>=** operator to find assets whose information was last updated the day before the search was started; or you can specify **now/w** with the **<=** operator to find assets whose information was last updated between the start of the first day of the current week (00:00:00:000 UTC) and the current time, **now** .

KUMA stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the period, the category will cover assets from 03:00:00.000 until now, not from 00:00:00.000 until now.

If you want to take your time zone into account when selecting a relative period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the date and time field by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want the categorization to cover the **Yesterday** period, you need to change the value to **now-1d/d-3h**. If you want the categorization to cover the **Today** period, change the value to **now/d-3h**.

Exporting asset data

You can export data about the assets displayed in the assets table as a CSV file.

To export asset data:

1. Configure the assets table.

   Only the data specified in the table is written to the file. The display order of the asset table columns is preserved in the exported file.

2. Find the desired assets and select the check boxes next to them.

   You can select all the assets in the table at a time by selecting the check box in the left part of the assets table header.

3. Click the **Export CSV** button.

The asset data is written to the assets_<export date>_<export time>.csv file. The file is downloaded according to your browser settings.

Viewing asset details

To view information about an asset, open the asset information window in one of the following ways:

- In the main menu, select the **Asset management** section → select a category with the relevant assets → select an asset.
- In the main menu, select **Alerts** → click the link with the relevant alert → select the asset in the **Related endpoints** section.
- In the main menu, select the **Events** section → search and filter events → select the relevant event → click the link in one of the following fields: SourceAssetID, DestinationAssetID, or DeviceAssetID.

The following information may be displayed in the asset details window:

- **Name**—asset name.

  Assets imported into OSMP retain the names that were assigned to them at the source. You can change these names in the OSMP web interface.

- **Tenant**—the name of the tenant that owns the asset.
- **Asset source**—source of information about the asset. There may be multiple sources. For instance, information can be added in the OSMO web interface or by using the API, or it can be imported from the Open Single Management Platform, KICS/KATA, or MaxPatrol reports.

  When using multiple sources to add information about the same asset to OSMP, you should take into account the rules for merging asset data.

- **Created**—date and time when the asset was added to OSMP.
- **Updated**—date and time when the asset information was most recently modified.
- **Owner**—owner of the asset, if provided.
- **IP address**—IP address of the asset (if any).

  > If there are several assets with identical IP addresses in OSMP, the asset that was added the latest is returned in all cases when assets are searched by IP address. If assets with identical IP addresses can coexist in your organization's network, plan accordingly and use additional attributes to identify the assets. For example, this may become important during correlation.

- **FQDN**—Fully Qualified Domain Name of the asset, if provided.
- **MAC address**—MAC address of the asset (if any).
- **Operating system**—operating system of the asset.
- **Related alerts**—alerts associated with the asset (if any).

  To view the list of alerts related to an asset, click the **Find in Alerts** link. This opens the **Alerts** tab with the search expression set to filter all assets with the corresponding asset ID.

- **Software info** and **Hardware info**—if the asset software and hardware parameters are provided, they are displayed in this section.

- Asset vulnerability information:

  - **Kaspersky Security Center vulnerabilities**—vulnerabilities of the asset, if provided. This information is available for the assets imported from the Open Single Management Platform.

    You can learn more about the vulnerability by clicking the icon, which opens the Kaspersky Threats portal. You can also update the vulnerabilities list by clicking the **Update** link and requesting updated information from Open Single Management Platform.

  - **KICS/KATA vulnerabilities**—vulnerabilities of the asset, if any. This information is available for the assets imported from KICS/KATA.

- Asset source information:

  - **Last visible**—time when information about the asset was last received from Open Single Management Platform. This information is available for the assets imported from the Open Single Management Platform.

  - **Host ID**—ID of the Open Single Management Platform *Network Agent* from which the asset information was received. This information is available for the assets imported from the Open Single Management Platform. This ID is used to determine the uniqueness of the asset in Open Single Management Platform.

  - **KICS/KATA server IP address** and **KICS/KATA connector ID**—data on the KICS/KATA instance from which the asset was imported.

- **Custom fields**—data written to the custom fields of assets.

- Additional information about the protection settings of an asset with Kaspersky Endpoint Security for Windows or Kaspersky Endpoint Security for Linux installed:

  - **KSC extended status ID** – asset status. It can have the following values:

    - OK

    - Critical

    - Warning

  - **KSC extended status** – information about the asset status. For example, **Anti-virus databases were updated too long ago**.

  - **Real-time protection status** – status of Kaspersky applications installed on the asset. For example: **Running (if anti-virus application does not support categories of state Running)**.

  - **Encryption status** – information about asset encryption. For example: **Host doesn't have any encryption rules**.

  - **Spam protection status** – status of anti-spam protection. For example, **Starting**.

  - **Anti-virus protection status of mail servers** – status of the virus protection of mail servers. For example, **Starting**.

  - **Data Leakage Prevention status** – status of data leak protection. For example, **Starting**.

  - **Endpoint Sensor status** – status of data leak protection. For example, **Starting**.

- **Anti-virus databases last updated** – the version of the downloaded anti-virus databases.
- **Protection last updated** – the time when the anti-virus databases were last updated.
- **System last started** – the time when the system was last started.

  This information is displayed if the asset was imported from Open Single Management Platform.

- **Categories**—categories associated with the asset (if any).
- **CII category**—information about whether an asset is a critical information infrastructure (CII) object.

By clicking the **Move to KSC group** button, you can move the asset that you are viewing between Open Single Management Platform administration groups. You can also click the **Start task** drop-down list to run tasks available on the asset:

- By clicking the **KSC response** button, you can start a Open Single Management Platform task on the asset.
- By clicking the **KEDR response** button, you can run a Kaspersky Endpoint Detection and Response task on the asset.
- By clicking the **Refresh KSC asset** button, you can run a task to refresh information about the asset from Open Single Management Platform.

The tasks are available when integrated with Open Single Management Platform and when integrated with Kaspersky Endpoint Detection and Response.

Adding assets

You can add asset information in the following ways:

- Manually.

  You can add an asset using OSMP Console or the API.

- Import assets.

  You can import assets **from Kaspersky Security Center**, KICS for Networks, and MaxPatrol reports.

  All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

  KUMA imports the following device information received from Kaspersky Security Center Network Agents:

  - Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
  - Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Click **Import**, and then select one of the following actions:
   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.
   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.
3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

   If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.
4. Click the **Import** button.

The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

When assets are added, assets that already exist in OSMP can be merged with the assets being added.

Asset merging algorithm:

1. Checking uniqueness of Kaspersky Security Center or KICS for Networks assets.
   - The uniqueness of an asset imported from Kaspersky Security Center is determined by the **Host ID** parameter, which contains the Kaspersky Security Center Network Agent *Network Agent* identifier. If two assets' IDs differ, they are considered to be separate assets and are not merged.
   - The uniqueness of an asset imported from KICS for Networks is determined by the combination of the **IP address**, **KICS for Networks server IP address**, and **KICS for Networks connector ID** parameters. If any of the parameters of two assets differ they are considered to be separate assets and are not merged.

   If the compared assets match, the algorithm is performed further.
2. Make sure that the values in the **IP**, **MAC**, and **FQDN** fields match.

   If at least two of the specified fields match, the assets are combined, provided that the other fields are blank.

   Possible matches:
   - The FQDN and IP address of the assets match. The **MAC** field is blank.

     The check is performed against the entire array of IP address values. If the IP address of an asset is included in the FQDN, the values are considered to match.
   - The FQDN and MAC address of the assets match. The **IP** field is blank.

     The check is performed against the entire array of MAC address values. If at least one value of the array fully matches the FQDN, the values are considered to match.

- The IP address and MAC address of the assets match. The **FQDN** field is blank.

  The check is performed against the entire array of IP- and MAC address values. If at least one value in the arrays is fully matched, the values are considered to match.

3. Make sure that the values of at least one of the **IP**, **MAC**, or **FQDN** fields match, provided that the other two fields are not filled in for one or both assets.

   Assets are merged if the values in the field match. For example, if the FQDN and IP address are specified for a KUMA asset, but only the IP address with the same value is specified for an imported asset, the fields match. In this case, the assets are merged.

   For each field, verification is performed separately and ends on the first match.

You can see examples of asset field comparison here.

Information about assets can be generated from various sources. If the added asset and the OSMP asset contain data received from the same source, this data is overwritten. For example, a Kaspersky Security Center asset receives a fully qualified domain name, software information, and host ID when imported into OSMP. When importing an asset from Kaspersky Security Center with an equivalent fully qualified domain name, all this data will be overwritten (if it has been defined for the added asset). All fields in which the data can be refreshed are listed in the Updatable data table.

Updatable data

Field name	Update procedure
Name	Selected according to the following priority: • Manually defined. • Received from Kaspersky Security Center. • Received by KICS for Networks.
Owner	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Manually defined.
IP address	The data is merged. If the array of addresses contains identical addresses, the copy of the duplicate address is deleted.
FQDN	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Received by KICS for Networks. • Manually defined.
MAC address	The data is merged. If the array of addresses contains identical addresses, one of the duplicate addresses is deleted.
Operating system	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Received by KICS for Networks. • Manually defined.
Vulnerabilities	KUMA asset data is supplemented with information from the added assets. In the asset details, data is grouped by the name of the source. Vulnerabilities are eliminated for each source separately.
Software info	Data from KICS for Networks is always recorded (if available). For other sources, the first value is selected according to the following priority: • Received from Kaspersky Security Center. • Manually defined.
Hardware info	The first value from the sources is selected according to the following priority: • Received from Kaspersky Security Center. • Defined via the API.

The updated data is displayed in the asset details. You can view asset details in OSMP Console.

This data may be overwritten when new assets are added. If the data used to generate asset information is not updated from sources for more than 30 days, the asset is deleted. The next time you add an asset from the same sources, a new asset is created.

If you are using OSMP Console to edit asset information that was received from Kaspersky Security Center or KICS for Networks, you can edit the following asset data:

- Name.
- Category.

If asset information was added manually, you can edit the following asset data when editing these assets in OSMP Console:

- Name.
- Name of the tenant that owns the asset.
- IP address.
- Fully qualified domain name.
- MAC address.
- Owner.
- Category.
- Operating system.
- Hardware info.

Asset data cannot be edited via the REST API. When importing from the REST API, the data is updated according to the rules for merging asset details provided above.

Adding asset information in OSMP

To add an asset in OSMP:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**, and then click **Add**.

   The **Add asset** details area opens in the right part of the window.

2. Enter the asset parameters:

   - **Asset name** (required)
   - **Tenant** (required)
   - **IP address** and/or **FQDN** (required). You can specify multiple FQDNs separated by commas
   - **MAC address**
   - **Owner**

3. If required, assign one or multiple categories to the asset:

   a. Click the button.

      Select categories window opens.

   b. Select the check boxes next to the categories that should be assigned to the asset. You can use the and icons to expand or collapse the lists of categories.

   c. Click **Save**.

   The selected categories appear in the **Categories** fields.

4. If required, add information about the operating system installed on the asset in the **Software** section.

5. If required, add information about asset hardware in the **Hardware info** section.

6. Click **Add**.

   The asset is created and displayed in the assets table in the category assigned to it or in the **Uncategorized assets** category.

Importing asset information from Kaspersky Security Center

All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

KUMA imports the following device information received from Kaspersky Security Center Network Agents:

- Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
- Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.

2. Click **Import**, and then select one of the following actions:

   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.

   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.

3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.

4. Click the **Import** button.

   The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

Importing asset information from MaxPatrol

You can import asset information from MaxPatrol network device scan reports into XDR. Imported assets are displayed in the **Assets** group. If necessary, you can edit the settings of assets.

You can import asset information either from a MaxPatrol report or from MaxPatrol VM.

Importing asset information from a MaxPatrol report

The import is performed through the API by using the maxpatrol-tool. The tool is located in the /opt/kaspersky/kuma/utils directory.

> Imports from MaxPatrol 8 are supported.

To import asset information from a MaxPatrol report:

1. In MaxPatrol, generate a network asset scan report in **XML file** format and copy the report file to the KUMA Core server. For more details about scan tasks and output file formats, refer to the MaxPatrol documentation.

   > Data cannot be imported from reports in **SIEM integration file** format. The **XML file** format must be selected.

2. Create a file with the token for accessing the KUMA REST API. For convenience, it is recommended to place it into the MaxPatrol report folder. The file must not contain anything except the token.

   Requirements imposed on accounts for which the API token is generated:

   - Administrator or Analyst role.
   - Access to the tenant into which the assets will be imported.
   - Permissions for using API requests GET /users/whoami and POST /api/v1/assets/import have been configured.

     > To import assets from MaxPatrol, it is recommended to create a separate user with the minimum necessary set of rights to use API requests.

3. Copy the maxpatrol-tool to the server hosting the KUMA Core and make the tool's file executable by running the following command:

   ```
   chmod +x <path to the maxpatrol-tool file on the server hosting the KUMA Core>
   ```

4. Run the maxpatrol-tool:

```
./maxpatrol-tool --kuma-rest <KUMA REST API server address and port> --token <path and name of API token file> --tenant <name of tenant where assets will reside> <path and name of MaxPatrol report file> --cert <path to the KUMA Core certificate file>
```

Example: `./maxpatrol-tool --kuma-rest example.kuma.com:7223 --token token.txt --tenant Main example.xml --cert /opt/kaspersky/kuma/core/certificates/ca.cert`

You can use additional flags and commands for import operations. For example, the command `--verbose, -v` will display a full report on the received assets. A detailed description of the available flags and commands is provided in the table titled Flags and commands of maxpatrol-tool. You can also use the `--help` command to view information on the available flags and commands.

The asset information will be imported from the MaxPatrol report to KUMA. The console displays information on the number of new and updated assets.

Example:
inserted 2 assets;
updated 1 asset;
errors occurred: []

The tool works as follows when importing assets:

- KUMA overwrites the data of assets imported through the API, and deletes information about their resolved vulnerabilities.
- KUMA skips assets with invalid data. Error information is displayed when using the `--verbose` flag.
- If there are assets with identical IP addresses and fully qualified domain names (FQDN) in the same MaxPatrol report, these assets are merged. The information about their vulnerabilities and software is also merged into one asset.

When uploading assets from MaxPatrol, assets that have equivalent IP addresses and fully qualified domain names (FQDN) that were previously imported from Kaspersky Security Center are overwritten.

To avoid this problem, you must configure range-based asset filtering by running the following command:

```
--ignore <IP address ranges> or -i <IP address ranges>
```

Assets that satisfy the filtering criteria are not uploaded. For a description of this command, please refer to the table titled *Flags and commands of maxpatrol-tool*.

Flags and commands of maxpatrol-tool

Flags and commands	Description
`--kuma-rest <KUMA REST API server port and address>, -a <KUMA REST API server port and address>`	Address (with the port) of KUMA Core server where assets will be imported. For example, `example.kuma.com:7223`. Port 7223 is used for API requests by default. You can change the port if necessary.
`--token <path and name of API token file>, -t <path and name of API token file>`	Path and name of the file containing the token used to access the REST API. This file must contain only the token. The Administrator or Analyst role must be assigned to the user account for which the API token is being generated.
`--tenant <tenant name>, -T <tenant name>`	Name of the KUMA tenant in which the assets from the MaxPatrol report will be imported.
`--dns <IP address ranges> or -d <IP address ranges>`	This command uses DNS to enrich IP addresses with FQDNs from the specified ranges if the FQDNs for these addresses were not already specified. Example: `--dns 0.0.0.0-9.255.255.255,11.0.0.0-255.255.255,10.0.0.2`

Flags and commands	Description
`--dns-server <DNS server IP address>, -s <DNS server IP address>`	Address of the DNS server that the tool must contact to receive FQDN information. Example: `--dns-server 8.8.8.8`
`--ignore <IP address ranges> or -i <IP address ranges>`	Address ranges of assets that should be skipped during import. Example: `--ignore 8.8.0.0-8.8.255.255, 10.10.0.1`
`--verbose, -v`	Output of the complete report on received assets and any errors that occurred during the import process.

`--help, -h`
`help`
	Get reference information on the tool or a command.
	Examples:
	`./maxpatrol-tool help`
	`./maxpatrol-tool < command > --help`

`version`	Get information about the version of the maxpatrol-tool.
`completion`	Creation of an autocompletion script for the specified shell.
`--cert <path to file with the KUMA Core certificate>`	Path to the KUMA Core certificate. By default, the certificate is located in the folder with the application installed: /opt/kaspersky/kuma/core/certificates/ca.cert.

Examples:

- `./maxpatrol-tool --kuma-rest example.kuma.com:7223 --token token.txt --tenant Main example.xml --cert /example-directory/ca.cert` – import assets to KUMA from MaxPatrol report example.xml.
- `./maxpatrol-tool help`—get reference information on the tool.

Possible errors

Error message	Description
must provide path to xml file to import assets	The path to the MaxPatrol report file was not specified.
incorrect IP address format	Invalid IP address format. This error may arise when incorrect IP ranges are indicated.
no tenants match specified name	No suitable tenants were found for the specified tenant name using the REST API.
unexpected number of tenants (%v) match specified name. Tenants are: %v	KUMA returned more than one tenant for the specified tenant name.
could not parse file due to error: %w	Error reading the XML file containing the MaxPatrol report.
error decoding token: %w	Error reading the API token file.
error when importing files to KUMA: %w	Error transferring asset information to KUMA.
skipped asset with no FQDN and IP address	One of the assets in the report did not have an FQDN or IP address. Information about this asset was not sent to KUMA.
skipped asset with invalid FQDN: %v	One of the assets in the report had an incorrect FQDN. Information about this asset was not sent to KUMA.
skipped asset with invalid IP address: %v	One of the assets in the report had an incorrect IP address. Information about this asset was not sent to KUMA.
KUMA response: %v	An error occurred with the specified report when importing asset information.
unexpected status code %v	An unexpected HTTP code was received when importing asset information from KUMA.

Importing asset information from MaxPatrol VM

The OSMP distribution kit includes the kuma-ptvm utility, which consists of an executable file and a configuration file. The utility is supported on Windows and Linux operating systems. The utility allows you to connect to the MaxPatrol VM API to get data about devices and their attributes, including vulnerabilities, and also lets you edit asset data and import data using the XDR API. Importing data is supported for MaxPatrol VM versions 1.1–2.7 (inclusive).

Configuring the import of asset information from MaxPatrol VM to KUMA Core involves the following steps:

1. Preparing XDR and MaxPatrol VM.

   You must create user accounts and an XDR token for API operations.

2. Creating a configuration file with data export and import settings.

3. Importing asset data into KUMA Core using the kuma-ptvm utility:

   a. The data is exported from MaxPatrol VM and saved in the directory of the utility. Information for each tenant is saved to a separate file in JSON format.

      If necessary, you can edit the received files.

   b. Information from files is imported into KUMA Core.

When re-importing existing assets, assets that already exist in KUMA Core are overwritten. In this way, fixed vulnerabilities are removed.

Known limitations:

- If the same IP address is specified for two assets with different FQDNs, KUMA Core imports such assets as two different assets; the assets are not combined.
- If an asset has two softwares with the same data in the name, version, vendor fields, KUMA Core imports this data as one software, despite the different software installation paths in the asset.
- If the FQDN of an asset contains a space or underscore ("_"), data for such assets is not imported into KUMA Core, and the log indicates that the assets were skipped during import.
- If an error occurs during import, error details are logged and the import stops.

Preparatory actions:

1. Create a separate user account in XDR and in MaxPatrol VM with the minimum necessary set of permissions to use API requests.

2. Create user accounts for which you will later generate an API token.

   Requirements imposed on accounts for which the API token is generated:

   - Main administrator, Tenant administrator, Tier 2 analyst, or Tier 1 analyst role.
   - Access to the tenant into which the assets will be imported.
   - In the user account, under API access rights, the check box is selected for POST /xdr/api/v2.1/kuma/assets/import.

3. Generate a token for access to the XDR REST API.

To create the configuration file:

1. Go to the KUMA utilities folder:

```
cd /opt/kaspersky/kuma/utils/
```

2. Copy the kuma-ptvm-config-template.yaml template to create a configuration file named kuma-ptvm-config.yaml:

```
cp kuma-ptvm-config-template.yaml kuma-ptvm-config.yaml
```

3. Edit the settings in the kuma-ptvm-config.yaml configuration file.

4. Save the changes to the file.

The configuration file will be created.

To import asset information:

1. If you want to import asset information from MaxPatrol VM into KUMA Core without intermediate verification of the exported data, run the kuma-ptvm utility with the following options:

   `kuma-ptvm --config <path to the kuma-ptvm-config.yaml file> --download --upload`

2. If you want to check the correctness of data exported from MaxPatrol VM before importing it into KUMA Core:

   a. Run the kuma-ptvm utility with the following options:

   `kuma-ptvm --config <path to the kuma-ptvm-config.yaml file> --download`

   For each tenant specified in the configuration file, a separate file is created with a name of the form <tenant ID>.JSON. Also, during export, a 'tenants' file is created, containing a list of JSON files to be uploaded to KUMA Core. All files are saved in the utility's directory.

   b. Review the exported asset files and if necessary, make the following edits:

   - Assign assets to their corresponding tenants.
   - Manually transfer asset data from the 'default' tenant file to the files of the relevant tenants.
   - In the 'tenants' file, edit the list of tenants whose assets you want to import into KUMA Core.

   c. Import asset information into KUMA Core:

   `kuma-ptvm --config <path to the kuma-ptvm-config.yaml file> --upload`

   To view information about the available commands of the utility, run the --help command.

   The asset information is imported from MaxPatrol VM to KUMA Core. The console displays information on the number of new and updated assets.

Possible errors:

When running the kuma-ptvm utility, the `tls: failed to verify certificate: x509: certificate is valid for localhost` error may be returned.

To resolve the issue:

- Issue a certificate in accordance with the MaxPatrol documentation. We recommend resolving the error in this way.
- Disable certificate validation.

  To disable certificate validation, add the following line to the configuration file in the 'MaxPatrol settings' section:

```
ignore_server_cert: true
```

As a result, the utility is started without errors.

Importing asset information from KICS for Networks

Information about assets protected by KICS for Networks can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

After configuring KICS for Networks integration, tasks to obtain data about KICS for Networks assets are created automatically. This occurs:

- Immediately after creating a new integration.
- Immediately after changing the settings of an existing integration.
- According to a regular schedule every several hours. Every 12 hours by default. The schedule can be changed.

Account data update tasks can be created manually.

To configure a task to update KICS for Networks asset data for a tenant:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant into which you want to import KICS/KATA assets.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **KICS/KATA import** tab, make sure that the **State** toggle switch is turned on.
5. Configure the asset data import:
   - To schedule regular import, in the **Import interval (hours)** field, specify the time interval at which OSMP updates data about KICS/KATA assets.
   - To run import manually, click the **Import** button.

   A **task** to receive account data from the selected tenant is added to the Task manager section of OSMP Console.

Examples of asset field comparison during import

Each imported asset is compared to the matching OSMP asset.

Checking for two-field value match in the IP, MAC, and FQDN fields

Compared assets	Compared fields		
	FQDN	IP	MAC
KUMA asset	Filled in	Filled in	Empty
Imported asset 1	Filled in, matching	Filled in, matching	Filled in
Imported asset 2	Filled in, matching	Filled in, matching	Empty
Imported asset 3	Filled in, matching	Empty	Filled in
Imported asset 4	Empty	Filled in, matching	Filled in
Imported asset 5	Filled in, matching	Empty	Empty
Imported asset 6	Empty	Empty	Filled in

Comparison results:

- Imported asset 1 and OSMP asset: the FQDN and IP fields are filled in and match, no conflict in the MAC fields between the two assets. The assets are merged.
- Imported asset 2 and OSMP asset: the FQDN and IP fields are filled in and match. The assets are merged.
- Imported asset 3 and OSMP asset: the FQDN and MAC fields are filled in and match, no conflict in the IP fields between the two assets. The assets are merged.
- Imported asset 4 and OSMP asset: the IP fields are filled in and match, no conflict in the FQDN and MAC fields between the two assets. The assets are merged.
- Imported asset 5 and OSMP asset: the FQDN fields are filled in and match, no conflict in the IP and MAC fields between the two assets. The assets are merged.
- Imported asset 6 and OSMP asset: no matching fields. The assets are not merged.

Checking for single-field value match in the IP, MAC, and FQDN fields

Compared assets	Compared fields		
	FQDN	IP	MAC
OSMP asset	Empty	Filled in	Empty
Imported asset 1	Filled in	Filled in, matching	Filled in
Imported asset 2	Filled in	Filled in, matching	Empty
Imported asset 3	Filled in	Empty	Filled in
Imported asset 4	Empty	Empty	Filled in

Comparison results:

- Imported asset 1 and OSMP asset: the IP fields are filled in and match, no conflict in the FQDN and MAC fields between the two assets. The assets are merged.
- Imported asset 2 and OSMP asset: the IP fields are filled in and match, no conflict in the FQDN and MAC fields between the two assets. The assets are merged.

- Imported asset 3 and OSMP asset: no matching fields. The assets are not merged.
- Imported asset 4 and OSMP asset: no matching fields. The assets are not merged.

Settings of the kuma-ptvm-config.yaml configuration file

The table lists the settings that you can specify in the kuma-ptvm-config.yaml file.

Setting	Description	Values
`log_level`	An optional setting in the 'General settings' group. Logging level.	Available values: • `trace` • `info` • `warning` • `error` Default setting: `info`.
`period`	An optional setting in the 'General settings' group. Data for assets that have changed during the specified period is exported from MaxPatrol.	No limitations apply. Default setting: 30d.
`strict_import`	Optional setting in the 'General settings' group. When exporting assets from MaxPatrol, check if the required fields for KUMA are filled. Do not export unverified assets from MaxPatrol.	Available values: • `true` to check for the presence of fields that are required for KUMA. • `false` to skip the check for the presence of fields that are required for KUMA. Default setting: `false`. We recommend specifying true when exporting assets from MaxPatrol, this lets you detect and fix possible errors in JSON files before you import assets into XDR.
`endpoint`	Required setting in the 'KUMA settings' group. URL of the XDR API server. For example, `api.<XDR FQDN>/xdr/`	-
`token`	Required setting in the 'KUMA settings' group. XDR API token.	-
`ignore_server_cert`	Optional setting in the 'KUMA settings' group. Validation of the XDR certificate.	Available values: `true` to disable certificate validation. `false` to enable certificate validation. This setting is not included in the configuration file template. You can manually add this setting with a true value, which will prevent the kuma-ptvm utility from validating the certificate at startup.
`endpoint`	Required setting in the 'MaxPatrol VM' group. URL of the MaxPatrol API server.	-
`user`	Required setting in the 'MaxPatrol VM' group. MaxPatrol API user name.	-
`password`	Required setting in the 'MaxPatrol VM' group. MaxPatrol API user password.	-
`secret`	Required setting in the 'MaxPatrol VM settings' group. MaxPatrol API secret.	-

`ignore_server_cert`	Optional setting in the 'MaxPatrol VM settings' group. Validation of the MaxPatrol certificate.	Available values: • `true` to disable the validation of the MaxPatrol certificate. • `false` to enable MaxPatrol certificate validation. This setting is not included in the configuration file template. You can manually add this setting with a `true` value if the `tls: failed to verify certificate: x509: certificate is valid for localhost` error occurs. In that case, the kuma-ptvm utility does not validate the certificate when it is started. We recommend issuing a certificate in accordance with the MaxPatrol documentation as the preferred way of resolving the error.
`only_exploitable`	Optional setting in the 'Vulnerability filter' group. Export from MaxPatrol only assets with vulnerabilities for which exploits are known.	Available values: `true` to export only assets with vulnerabilities for which exploits are known. `false` to export all assets. Default setting: `false`.
`min_severity`	Optional setting in the 'Vulnerability filter' group. Import only vulnerabilities of the specified level or higher.	Available values: • `low` • `medium` • `high` • `critical` Default value: `low`.
`id`	Required setting in the 'Tenant map' group. Tenant ID in XDR. Assets are assigned to tenants in the order in which tenants are specified in the configuration file: the higher a tenant is in the list, the higher its priority. This means you can specify overlapping subnets.	-
`fqdn`	Optional setting in the 'Tenant map' group. Regular expression for searching the FQDN of an asset.	-
`networks`	Optional setting in the 'Tenant map' group. One or more subnets.	-
`default_tenant`	Optional setting. The default XDR tenant for data about assets that could not be allocated to tenants specified in the 'Tenants' group of settings.	-

Assigning a category to an asset

To assign a category to one asset:

1. In the main menu, go to the **Asset management** section.
2. Select the category with the relevant assets.
   The assets table is displayed.
3. Select an asset.
4. In the opened window, click the **Edit** button.
5. In the **Categories** field, click the button.
6. Select a category.

   If you want to move an asset to the **Uncategorized assets** section, you must delete the existing categories for the asset by clicking the button.

7. Click the **Save** button.

The category will be assigned.

To assign a category to multiple assets:

1. In the main menu, go to the **Assets** section.
2. Select the category with the relevant assets.
   The assets table is displayed.
3. Select the check boxes next to the assets for which you want to change the category.
4. Click the **Link to category** button.
5. In the opened window, select a category.
6. Click the **Save** button.

The category will be assigned.

Do not assign the `Categorized assets` category to assets.

Linking a group of assets to a category

To link a group of assets to a category:

1. In the **Asset management** section, select the check box in the header of the table of assets.
2. Select all assets visible on the page or all assets that match the selection condition.

   The **Link to category** button becomes active and opens the available categories.
3. Click the **Link to category** button and select one or more categories to link to.
4. Click **OK**.

   Assets are linked to the selected categories or folder.

Unlinking a group of assets from a category

To unlink a group of assets from a category:

1. Select one category (tenant) in the navigation pane.

   The list of assets in that category is displayed.

   The **Clean category** button is added to the folder properties.
2. In the context menu of the category, select **Clean category**.

   A dialog box is displayed with a confirmation prompt and the number of assets that will be unlinked.

   This option lets you unlink all assets in the selected category, not only those that are visible on the page. Assets in child categories are not unlinked.
3. In the dialog box, click **OK**.

   All selected assets aer unlinked from the selected category.

Editing the parameters of assets

In OSMP, you can edit asset parameters. All the parameters of manually added assets can be edited. For assets imported from Kaspersky Security Center, you can only change the name of the asset and its category.

To change the parameters of an asset:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Select the category with the relevant assets.

   The assets table is displayed.
3. Select an asset, and then click **Details**.
4. In the opened window, click the **Edit** button.

   The **Edit asset** window opens.

5. Make the changes you need in the available fields:

- **Asset name** (required). This is the only field available for editing if the asset was imported from Kaspersky Security Center or KICS for Networks.
- **IP address** and/or **FQDN** (required). You can specify multiple FQDNs separated by commas.
- **MAC address**
- **Owner**
- **Software info**:
  - **OS name**
  - **OS build**
- **Hardware info**:

  Hardware parameters

  You can add information about asset hardware to the **Hardware info** section:

  Available fields for describing the asset CPU:

  - **CPU name**
  - **CPU frequency**
  - **CPU core count**

    You can add CPUs to the asset by using the **Add CPU** link.

  Available fields for describing the asset disk:

  - **Disk free bytes**
  - **Disk volume**

    You can add disks to the asset by using the **Add disk** link.

  Available fields for describing the asset RAM:

  - **RAM frequency**
  - **RAM total bytes**

  Available fields for describing the asset network card:

  - **Network card name**
  - **Network card manufacture**
  - **Network card driver version**

You can add network cards to the asset by using the **Add network card** link.

- Custom fields.
- CII category.

6. Assign or change the category of the asset:

   a. Click the button.

   Select categories window opens.

   b. Select the check boxes next to the categories that should be assigned to the asset.

   c. Click **Save**.

   The selected categories appear in the **Categories** fields.

   You can also select the asset and then drag and drop it into the relevant category. This category will be added to the list of asset categories.

   Do not assign the `Categorized assets` category to assets.

7. Click the **Save** button.

Asset parameters have been changed.

Archiving assets

In OSMP, the archival functionality is available for the following types of assets:

- For assets imported from Kaspersky Security Center and KICS.

  If OSMP did not receive information about the asset, at the time of import, the asset is automatically archived and is stored in the database for the time specified in the **TTL for archived assets (days)** setting. The default setting is 0 days. This means that archived assets are stored indefinitely. An archived asset becomes active if OSMP receives information about the asset from the source before the retention period for archived assets expires.

- Combined assets

  When importing, OSMP performs a check for uniqueness among assets imported from Kaspersky Security Center and KICS, and among manually added assets. If the fields of an imported asset and a manually added asset match, the assets are combined into a single asset, which is considered imported and can become archived.

Assets added manually in the console or using the API are not archived.

An asset becomes archived under the following conditions:

- OSMP did not receive information about the asset from Kaspersky Security Center or KICS for Networks.
- Disabled integration with Kaspersky Security Center.

  If you disable integration with Kaspersky Security Center, the asset is considered active for 30 days. After 30 days, the asset is automatically archived and is stored in the database for the time specified in the **TTL for archived assets (days)** field.

An asset is not updated in the following cases:

- Information about the Kaspersky Security Center asset has not been updated for more than the retention period of archived assets.
- Information about the asset does not exist in Kaspersky Security Center or KICS for Networks.
- Connection with the Kaspersky Security Center Server has not been established for more than 30 days.

To configure the archived assets retention period:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant for which you want to change the archived assets retention period.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **General** tab, enter the new value in the **TTL for archived assets (days)** field.

   The default setting is 0 days. This means that archived assets are stored indefinitely.
5. Click **Save**.

The retention period for archived assets is configured.

Information about the archived asset remains available for viewing in the alert and incident card.

To view an archived asset card:

1. In the main menu, under **Monitoring**, select the **Alerts** or **Incidents** section.

   A list of alerts or incidents is displayed.
2. Open the alert or incident card linked to the archived asset.

   You can view the information in the archived asset card.

Deleting assets

You can delete assets on the **Administration and protection** tab. On the **Analysis and response** tab, OSMP deletes only archived assets and only automatically, you cannot delete the assets manually. OSMP deletes an archived asset if the information about the asset has not been updated for longer than the retention period of archived assets.

Bulk deletion of assets

In the OSMP web interface, you can select multiple assets using a filter and delete all selected assets.

To delete assets, you must have rights to delete assets.

Bulk deletion of assets

To delete all selected assets:

1. In the main window of the application, go to the **Asset management** section and select an asset category.

   A table of assets is displayed.

2. Select the check box in the heading of the table of assets.

   You can delete all assets or all assets currently displayed on the page.

3. Click **Select all in page** or **Select all**.

   The **Delete** button becomes active.

4. Click the **Delete** button.

   This opens a window prompting you to confirm deletion and telling you that deleted assets will not be available in alerts, incidents, and widgets.

   In the lower part of the page, the number of assets selected for deletion is displayed.

5. Click **OK**.

   If you clicked **Select all**, you must enter the displayed generated string into the text box in the window to confirm deletion.

All selected assets are deleted.

Deleting asset folders

To delete a folder, you can either unlink all assets from the folder (which unlinks the assets but not does not delete them) and then delete the folder itself, or delete all assets and then delete the folder.

Updating third-party applications and fixing vulnerabilities on Kaspersky Security Center assets

You can update third-party applications (including Microsoft applications) that are installed on Kaspersky Security Center assets, and fix vulnerabilities in these applications.

First you need to create the *Install required updates and fix vulnerabilities* task on the selected Kaspersky Security Center Administration Server with the following settings:

- Application—Kaspersky Security Center.
- Task type—*Install required updates and fix vulnerabilities*.
- Devices to which the task will be assigned—you need to assign the task to the root administration group.
- Rules for installing updates:
  - Install approved updates only.
  - Fix vulnerabilities with a severity level equal to or higher than (optional setting).

    If this setting is enabled, updates fix only those vulnerabilities for which the severity level set by Kaspersky is equal to or higher than the value selected in the list (*Medium*, *High*, or *Critical*). Vulnerabilities with a severity level lower than the selected value are not fixed.
- Scheduled start—the task run schedule.

For details on how to create a task, please refer to the *Kaspersky Security Center Help Guide*.

> The *Install required updates and fix vulnerabilities* task is available with a Vulnerability and Patch Management license.

Next, you need to install updates for third-party applications and fix vulnerabilities on assets in OSMP.

To install updates and fix vulnerabilities in third-party applications on an asset in OSMP:

1. Open the asset details window in one of the following ways:
   - In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**, select a category with the relevant assets, then select an asset, and then click **Details**.
   - In the main menu, go to **Monitoring** → **Threat hunting**, then search and filter events, select the relevant event, and then click the link in one of the following fields: SourceAssetID, DestinationAssetID, or DeviceAssetID.
2. In the asset details window, expand the list of **Kaspersky Security Center vulnerabilities**.
3. Select the check boxes next to the applications that you want to update.
4. Click the **Upload updates** link.
5. In the opened window, select the check box next to the ID of the vulnerability that you want to fix.
6. If **No** is displayed in the **EULA accepted** column for the selected ID, click the **Approve updates** button.
7. Click the link in the **EULA URL** column and carefully read the text of the End User License Agreement.
8. If you agree to it, click **Accept selected EULAs**.

   The ID of the vulnerability for which the EULA was accepted shows **Yes** in the **EULA accepted successfully** column.

9. Repeat steps 7–10 for each required vulnerability ID.

10. Click **OK**.

    Updates will be uploaded and installed on the assets managed by the Administration Server where the task was started, and on the assets of all secondary Administration Servers.

    The terms of the End User License Agreement for updates and vulnerability patches must be accepted on each secondary Administration Server separately.

Updates are installed on assets where the vulnerability was detected.

You can update the list of vulnerabilities for an asset in the asset details window by clicking the **Update** link.

Moving assets to a selected administration group

You can move assets to a selected administration group of Kaspersky Security Center. In this case, the group policies and tasks will be applied to the assets. For more details on Kaspersky Security Center tasks and policies, please refer to the *Kaspersky Security Center Help Guide*.

Administration groups are added to OSMP when the hierarchy is loaded during **import of assets from Kaspersky Security Center**. First, you need to configure OSMP integration with Kaspersky Security Center.

All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

KUMA imports the following device information received from Kaspersky Security Center Network Agents:

- Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
- Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Click **Import**, and then select one of the following actions:
   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.
   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.
3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.

4. Click the **Import** button.

   The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

To move an asset to a selected administration group:

1. Open the asset details window in one of the following ways:

   - In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**, select a category with the relevant assets, then select an asset, and then click **Details**.

   - In the main menu, go to **Monitoring** → **Alerts**, click the link with the relevant alert, and then select the asset in the **Assets** section, and then click **View properties**.

2. In the asset details window, click **Response** → **Move to group**.

3. Select the group in the opened window.

   The selected group must be owned by the same tenant as the asset.

4. Click the **Save** button.

   The selected asset will be moved.

To move multiple assets to a selected administration group:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.

2. Select the category with the relevant assets.

3. Select the check boxes next to the assets that you want to move to the group.

4. Click click **Response** → **Move to group**.

   The button is active if all selected assets belong to the same Administration Server.

5. Select the group in the opened window.

6. Click the **Save** button.

   The selected assets will be moved.

You can see the specific group of an asset in the asset details.

Kaspersky Security Center assets information is updated in OSMP when information about assets is imported from Kaspersky Security Center. This means that a situation may arise when assets have been moved between administration groups in Kaspersky Security Center, but this information is not yet displayed in OSMP. When an attempt is made to move such an asset to an administration group in which it is already located, OSMP returns the **Failed to move assets to another KSC group** error.

Asset audit

OSMP can be configured to generate asset audit events under the following conditions:

- Asset was added to OSMP. The application monitors manual asset creation, as well as creation during import via the REST API and during import from **Kaspersky Security Center** or KICS for Networks.

  All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

  KUMA imports the following device information received from Kaspersky Security Center Network Agents:

  - Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
  - Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

  To import information about the Kaspersky Security Center assets:

  1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
  2. Click **Import**, and then select one of the following actions:
     - To import information about the basic attributes of assets, click **Import KSC assets**.

       This opens the **Import KSC assets** window.
     - To import information about other attributes of assets, click **Import KSC assets attributes**.

       This opens the **Import KSC assets attributes** window.
  3. In the drop-down list, select the tenant for which you want to perform the import.

     In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

     If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.
  4. Click the **Import** button.

  The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

- Asset parameters have been changed. A change in the value of the following asset fields is monitored:
  - Name
  - IP address
  - MAC address
  - FQDN
  - Operating system

  Fields may be changed when an asset is updated during import.

- Asset was deleted from OSMP. The program monitors manual deletion of assets, as well as automatic deletion of assets imported from Kaspersky Security Center and KICS for Networks, whose data is no longer being received.

- Vulnerability info was added to the asset. The program monitors the appearance of new vulnerability data for assets. Information about vulnerabilities can be added to an asset, for example, when importing assets from Kaspersky Security Center or KICS for Networks.

- Asset vulnerability was resolved. The program monitors the removal of vulnerability information from an asset. A vulnerability is considered to be resolved if data about this vulnerability is no longer received from any sources from which information about its occurrence was previously obtained.

- Asset was added to a category. The program monitors the assignment of an asset category to an asset.

- Asset was removed from a category. The program monitors the deletion of an asset from an asset category.

By default, if asset audit is enabled, under the conditions described above, OSMP creates not only audit events (`Type = 4`), but also base events (`Type = 1`).

Asset audit events can be sent to storage or to correlators, for example.

Configuring an asset audit

To configure an asset audit:

1. In the main menu, go to **Settings** → **Tenants**.

2. Select the tenant for which you want to configure an asset audit.

   This opens the tenant properties window.

   If you want to configure an asset audit for a new tenant, then first create the tenant.

3. On the **Settings** tab, under **Detection and response**, click **Assets audit**.

4. For each condition for generating asset audit events, select the destination to where the created events will be sent:

   a. In the settings block of the relevant type of asset audit events, use the **Add destination** drop-down list to select the type of destination to which the created events should be sent:

      - Select **Storage** if you want events to be sent to storage.
      - Select **Correlator** if you want events to be sent to the correlator.
      - Select **Other** if you want to select a different destination.

      > This type of resource includes correlator and storage services that were created in previous versions of the program.

      In the **Add destination** window that opens you must define the settings for event forwarding.

   b. Use the **Destination** drop-down list to select an existing destination or select **Create** if you want to create a new destination.

      If you are creating a new destination, fill in the settings as indicated in the destination description.

   c. Click **Save**.

   A destination has been added to the condition for generating asset audit events. Multiple destinations can be added for each condition.

5. Click **Save**.

   The asset audit has been configured. Asset audit events will be generated for those conditions for which destinations have been added. Click **Save**.

Storing and searching asset audit events

Asset audit events are considered to be base events and do not replace audit events. Asset audit events can be searched based on the following parameters:

Event field	Value
DeviceVendor	`Kaspersky`
DeviceProduct	`KUMA`
DeviceEventCategory	`Audit assets`

Enabling and disabling an asset audit

You can enable or disable asset audit for a tenant.

To enable or disable an asset audit for a tenant:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant for which you want to enable or disable an asset audit.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets audit**.
4. Turn on or turn off the **Audit events collection** toggle switch.
5. Click **Save**.

By default, when asset audit is enabled, when an audit condition occurs, two types of events are simultaneously created: a base event and an audit event.

You can disable the generation of base events with audit events.

To enable or disable the creation of base events for an individual condition:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant for which you want to enable or disable a condition for generating asset audit events.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets audit**.
4. Click a condition action name to enable or disable creation of the corresponding base events.
5. Click **Save**.

For disabled conditions, only audit events are created, and base events are not created.

Custom asset fields

In addition to the existing fields of the asset data model, you can create custom asset fields. Data from the custom asset fields is displayed when you view information about the asset. Custom fields can be filled in with data either manually or using the API.

You can create or edit the custom fields in OSMP in the **Assets** section, in the **Custom fields** table. The table has the following columns:

- **Name** – the name of the custom field that is displayed when you view information about the asset.
- **Default value** – the value that is written to the custom field when an asset is added to KUMA.
- **Mask** – a regular expression to which the value in the custom field must match.

To create a custom asset field:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant for which you want to change the archived assets retention period.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **General** tab, under **Custom fields** field, click **Add** button.

   An empty row is added to the **Custom fields** table. You can add multiple rows with the custom field settings at once.
5. Fill in the columns with the settings of the custom field:
   - **Name** (required)—from 1 to 128 characters in Unicode encoding.
   - **Default value**—from 1 to 1,024 Unicode characters.
   - **Mask**—from 1 to 1,024 Unicode characters.
6. Click **Save**.

A custom field is added to the asset data model.

To delete or edit a custom asset field:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the tenant for which you want to change the archived assets retention period.

   This opens the tenant properties window.
3. On the **Settings** tab, under **Detection and response**, click **Assets settings**.
4. On the **General** tab, make the necessary changes in the **Custom fields** table:
   - To delete a custom field, select the check box next to the required field, and then click the **Delete** button. Deleting a field also deletes the data written in this field for all assets.
   - You can change the values of the field settings. Changing the default value does not affect the data written in the asset fields before.
   - To change the display order of the fields, use the **Move up** and **Move down** buttons.
5. Click **Save**.

The changes are made.

Critical information infrastructure assets

In OSMP, you can tag assets related to the critical information infrastructure (CII) of the Russian Federation. This allows you to restrict the OSMP users capabilities to handle alerts and incidents, which are associated with the assets related to the CII objects.

> You can assign the CII category to assets if the license with the GosSOPKA module is active in OSMP.

Main administrators and users with the **Access to CII facilities** check box selected in their profiles can assign the CII category to an asset. If none of these conditions are met, the following restrictions apply to the user:

- The **CII category** group of settings is not displayed in the **Asset details** and **Edit asset** windows. You cannot view or change the CII category of an asset.
- Alerts and incidents associated with the assets of the CII category are not available for viewing. You cannot perform any actions on such alerts and incidents; they are not displayed in the table of alerts and incidents.
- The **CII** column is not displayed in the Alerts and Incidents tables.
- Search and closing of the alerts using the REST API is not available.

The CII category of an asset is displayed in the **Asset details** window in the **CII category** group of settings.

To change the CII category of an asset:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Select the category with the relevant assets.

   The assets table is displayed.
3. Select an asset, and then click **Details**.

   The **Asset details** window opens.
4. Click the **Edit** button and select one of the available values in the drop-down list:
   - **Information resource is not a CII object** – default value, indicating that the asset does not have a CII category. The users with the **Access to CII facilities** check box cleared in their profiles can work with such assets and the alerts and incidents related to these assets.
   - **CII object without importance category**.
   - **CII object of the third importance category**.
   - **CII object of the second importance category**.
   - **CII object of the first importance category**.
5. Click **Save**.

Deploying Kaspersky applications

This section describes Kaspersky applications deployment on client devices in your organization by means of OSMP Console.

Scenario: Kaspersky applications deployment

This scenario explains how to deploy Kaspersky applications through OSMP Console. You can use the Protection deployment wizard, or you can complete all necessary steps manually.

Stages

Kaspersky applications deployment proceeds in stages:

1. **Downloading and creating installation packages**

   Download the package manually.

   If you cannot install Kaspersky applications by means of Open Single Management Platform on some devices, for example, on remote employees' devices, you can create stand-alone installation packages for applications. If you use stand-alone packages to install Kaspersky applications, you do not have to create and run a remote installation task, nor create and configure tasks for Kaspersky Endpoint Security for Windows.

   Alternatively, you can download the distribution packages for Network Agent and security applications from the Kaspersky website. If the remote installation of the applications is not possible for some reason, you can use the downloaded distribution packages to install the applications locally.

2. **Creating, configuring, and running the remote installation task**

   This step is part of the Protection deployment wizard. If you choose not to run the Protection deployment wizard, you must create this task manually and configure it manually.

   You also can manually create several remote installation tasks for different administration groups or different device selections. You can deploy different versions of one application in these tasks.

   Make sure that all the devices on your network are discovered; then run the remote installation task (or tasks).

   > If you want to install Network Agent on devices with the SUSE Linux Enterprise Server 15 operating system, install the insserv-compat package first to configure Network Agent.

3. **Creating and configuring tasks**

   The *Update* task of Kaspersky Endpoint Security must be configured.

   Create this task manually and configure it manually. Make sure that the schedule for the task meets your requirements. (By default, the scheduled start for the task is set to **Manually**, but you might want to choose another option.)

4. **Creating policies**

   Create the policy for Kaspersky Endpoint Security manually. You can use the default settings of the policy; you can also modify the default settings of the policy according to your needs at any time.

5. **Verifying the results**

   Make sure that deployment was completed successfully: you have policies and tasks for each application, and these applications are installed on the managed devices.

Results

Completion of the scenario yields the following:

- All required policies and tasks for the selected applications are created.
- The schedules of tasks are configured according to your needs.
- The selected applications are deployed, or scheduled to be deployed, on the selected client devices.

Protection deployment wizard

To install Kaspersky applications, you can use the Protection deployment wizard. The Protection deployment wizard enables remote installation of applications either through specially created installation packages or directly from a distribution package.

The Protection deployment wizard performs the following actions:

- Downloads an installation package for application installation (if it was not created earlier). The installation package is located at **Discovery & deployment** → **Deployment & assignment** → **Installation packages**. You can use this installation package for the application installation in the future.
- Creates and runs a remote installation task for specific devices or for an administration group. The newly created remote installation task is stored in the **Tasks** section. You can later start this task manually. The task type is **Install application remotely**.

If you want to install Network Agent on devices with the SUSE Linux Enterprise Server 15 operating system, install the insserv-compat package first to configure Network Agent.

Step 1. Starting Protection deployment wizard

You can start the Protection deployment wizard manually at any time.

To start the Protection deployment wizard manually,

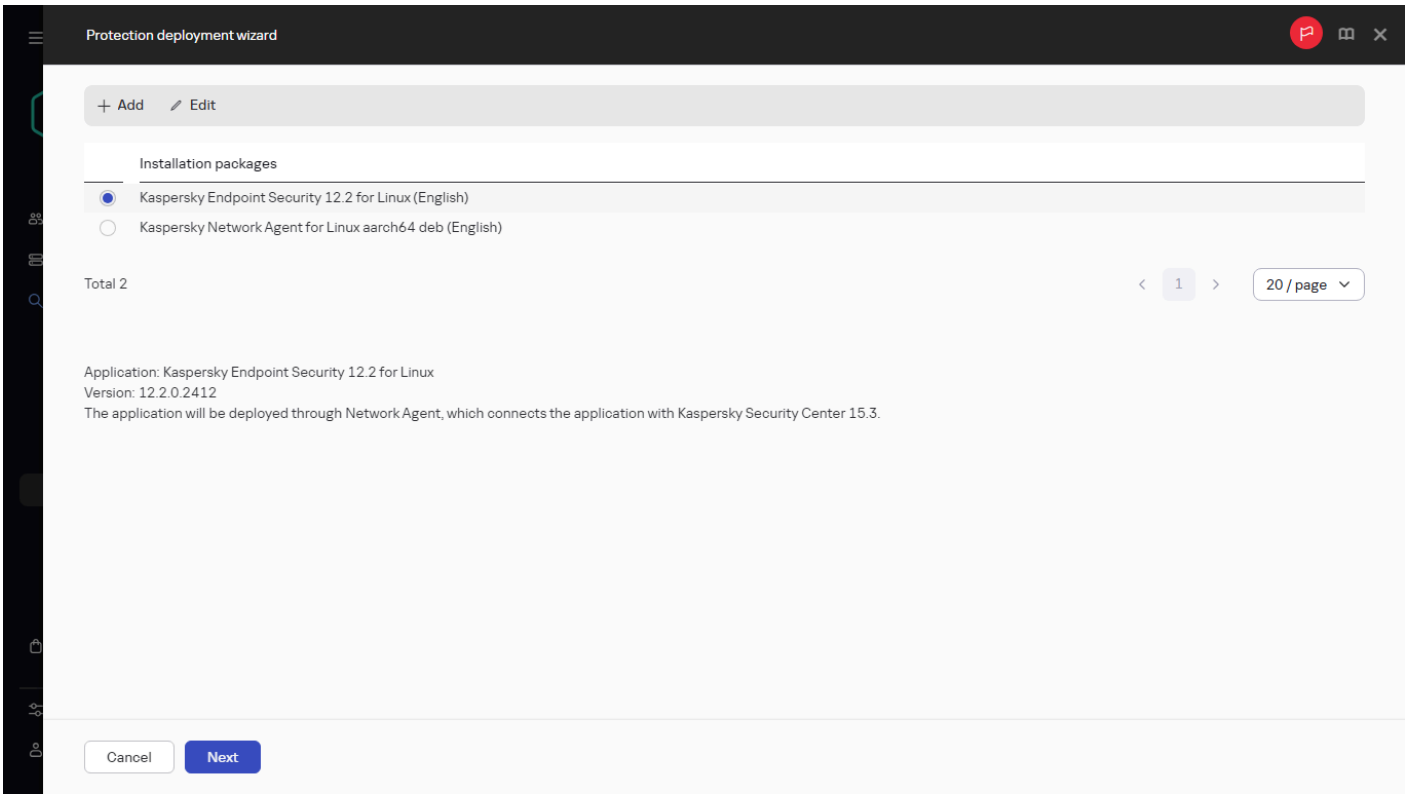
In the main menu, go to **Discovery & deployment** → **Deployment & assignment** → **Protection deployment wizard**.

The Protection deployment wizard starts. Proceed through the wizard by using the **Next** button.

Step 2. Selecting the installation package

Select the installation package of the application that you want to install.

If the installation package of the required application is not listed, click the **Add** button and then select the application from the list.



Select the installation package

Step 3. Selecting a method for distribution of key file or activation code

Select a method for the distribution of the key file or the activation code:

- **Do not add license key to installation package**

  The key is automatically distributed to all devices with which it is compatible:

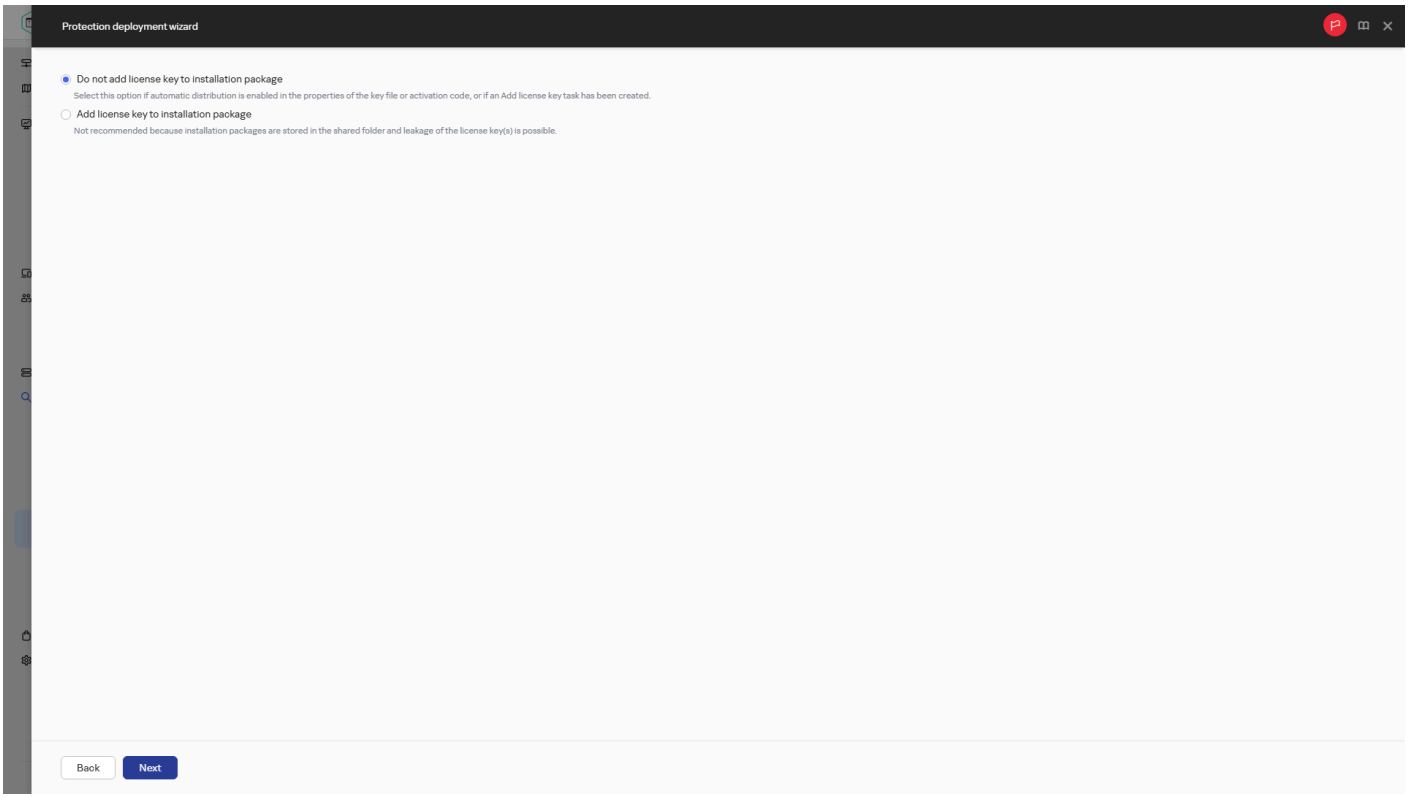
  - If automatic distribution has been enabled in the key properties.
  - If the **Add key** task has been created.

- **Add license key to installation package**

  The key is distributed to devices together with the installation package.

  We do not recommend that you distribute the key using this method because the shared Read access rights are enabled to the repository of installation packages.

If the installation package already includes a key file or an activation code, this window is displayed, but it only contains the license key information.

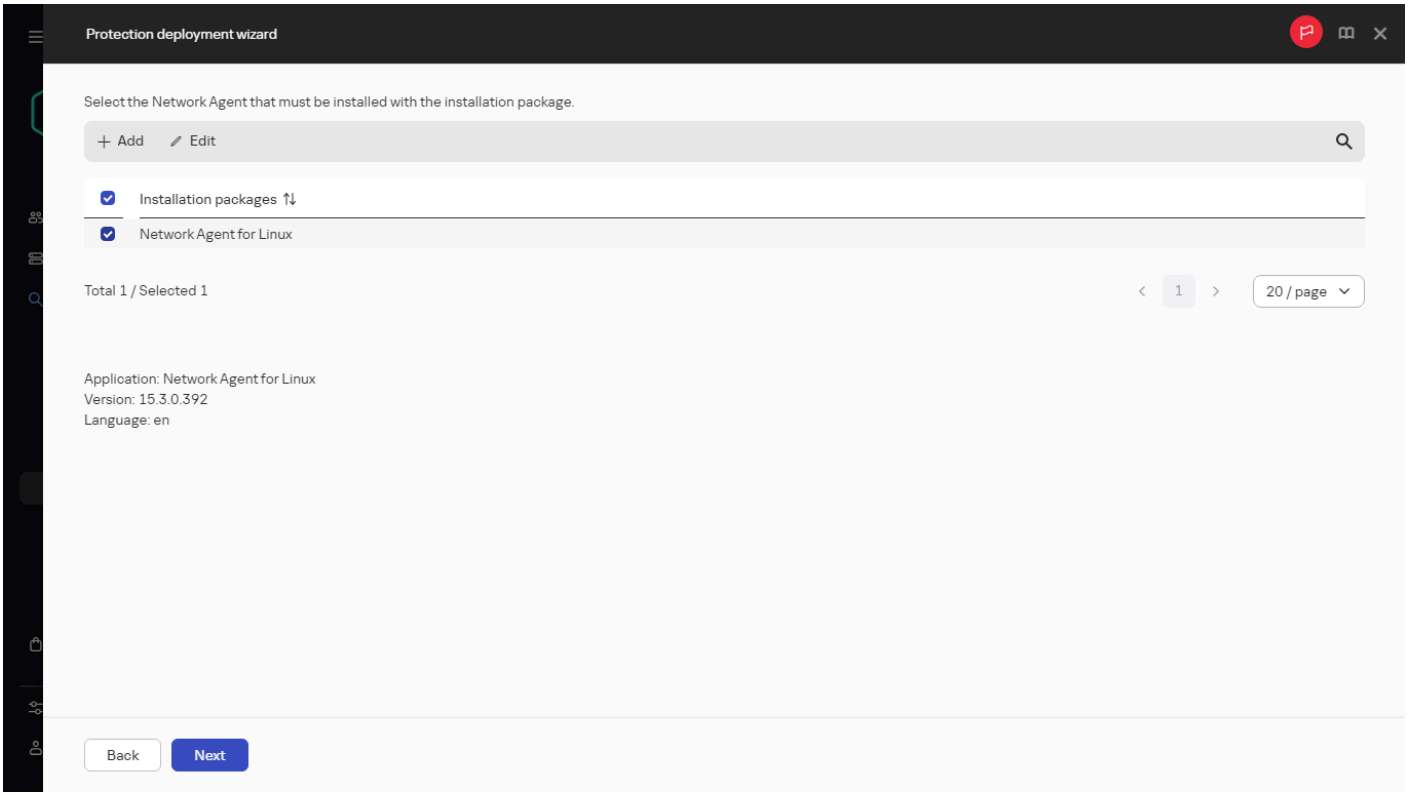


Selecting the distribution method of the key file or activation code

Step 4. Selecting Network Agent version

If you selected the installation package of an application other than Network Agent, you also have to install Network Agent, which connects the application with Kaspersky Security Center Administration Server.

Select the latest version of Network Agent.



Selecting the Network Agent version

Step 5. Selecting devices

Specify a list of devices on which the application will be installed:

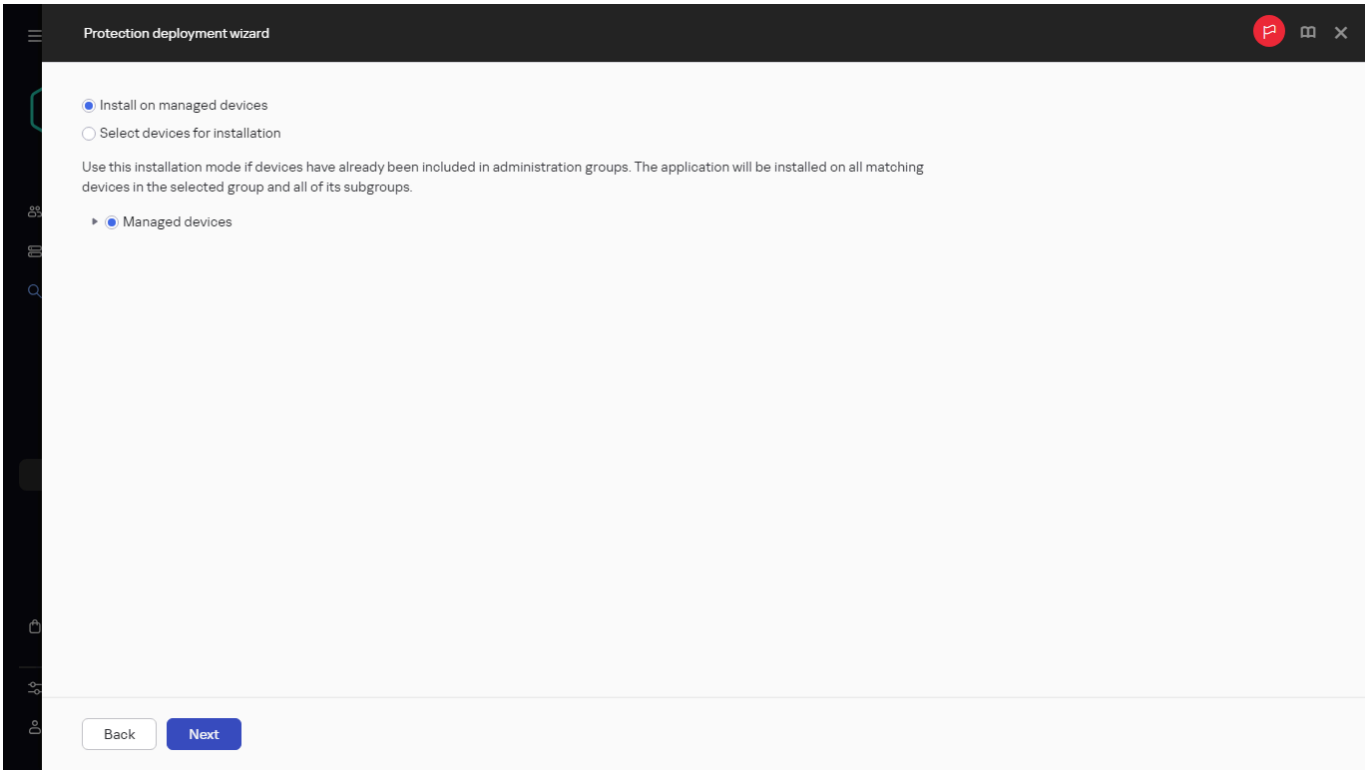
- **Install on managed devices**

  If this option is selected, the remote installation task is created for a group of devices.

- **Select devices for installation**

  The task is assigned to devices included in a device selection. You can specify one of the existing selections.

For example, you may want to use this option to run a task on devices with a specific operating system version.



Specifying the list of devices on which the application will be installed

Step 6. Specifying the remote installation task settings

On the **Remote installation task settings** page, specify the settings for remote installation of the application.

In the **Force installation package download** settings group, specify how files that are required for the application installation are distributed to client devices:

- **Using Network Agent**

  If this option is enabled, installation packages are delivered to client devices by Network Agent installed on those client devices.

  If this option is disabled, installation packages are delivered using the operating system tools of client devices.

  We recommend that you enable this option if the task has been assigned to devices with Network Agents installed.

  By default, this option is enabled.

- **Using operating system resources through distribution points**

  If this option is enabled, installation packages are transmitted to client devices using operating system tools through distribution points. You can select this option if there is at least one distribution point on the network.

  If the **Using Network Agent** option is enabled, the files are delivered using operating system tools only if Network Agent tools are unavailable.

By default, this option is enabled for remote installation tasks that have been created on a virtual Administration Server.

The only way to install an application for Windows (including Network Agent for Windows) on a device that does not have Network Agent installed is by using a Windows-based distribution point. Therefore, when you install a Windows application:

- Select this option.
- Ensure that a distribution point is assigned for the target client devices.
- Ensure the distribution point is Windows-based.

- **Using operating system resources through Administration Server**

  If this option is enabled, files are transmitted to client devices by using operating system tools of client devices through the Administration Server. You can enable this option if no Network Agent is installed on the client device, but the client device is in the same network as the Administration Server.

  By default, this option is enabled.

Define the additional setting:

- **Do not re-install application if it is already installed**

  If this option is enabled, the selected application will not be re-installed if it has already been installed on this client device.

  If this option is disabled, the application will be installed anyway.

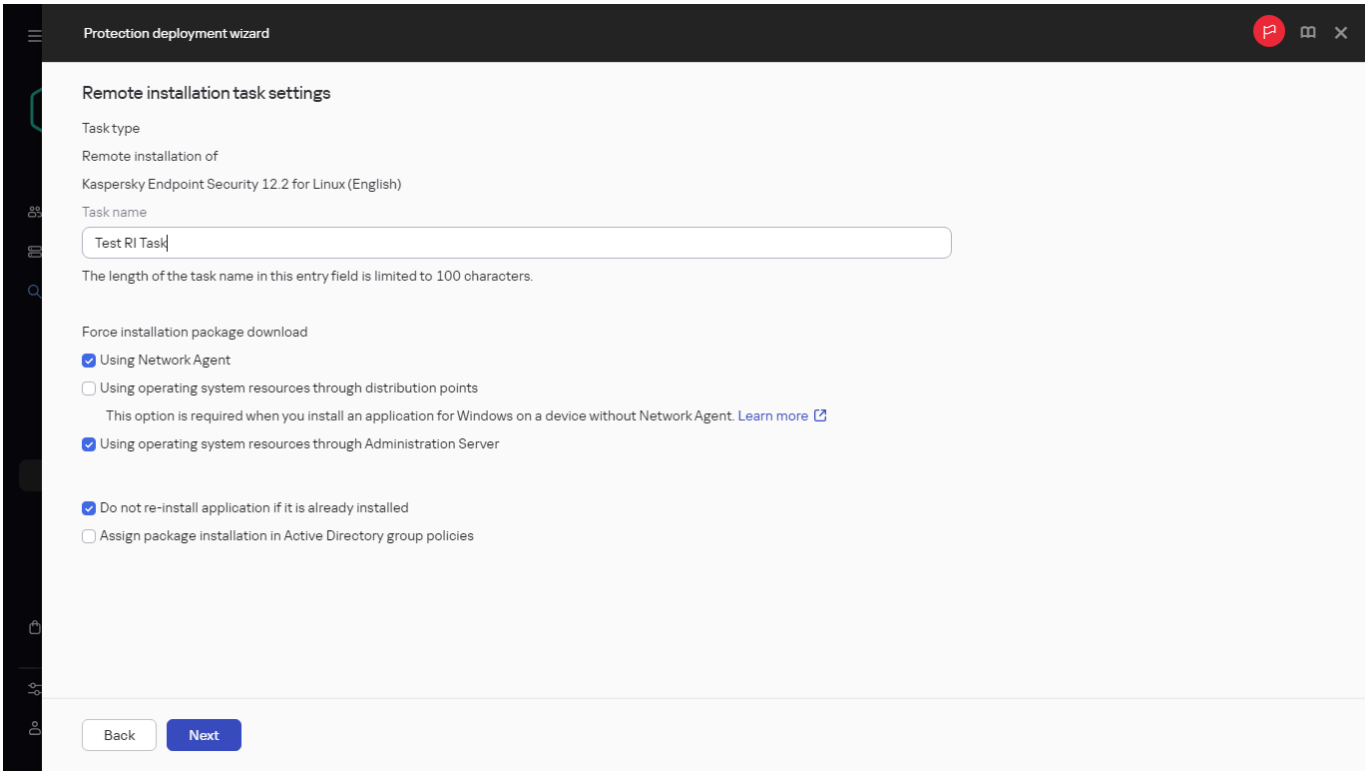
  By default, this option is enabled.

- **Assign package installation in Active Directory group policies**

  If this option is enabled, an installation package is installed by using the Active Directory group policies.

  This option is available if the Network Agent installation package is selected.

By default, this option is disabled.



Specifying the remote installation task settings

Step 7. Restart management

Specify the action to be performed if the operating system must be restarted when you install the application:

- **Do not restart the device**

  Client devices are not restarted automatically after the operation. To complete the operation, you must restart a device (for example, manually or through a device management task). Information about the required restart is saved in the task results and in the device status. This option is suitable for tasks on servers and other devices where continuous operation is critical.

- **Restart the device**

  Client devices are always restarted automatically if a restart is required for completion of the operation. This option is useful for tasks on devices that provide for regular pauses in their operation (shutdown or restart).

- **Prompt user for action**

  The restart reminder is displayed on the screen of the client device, prompting the user to restart it manually. Some advanced settings can be defined for this option: text of the message for the user, the message display frequency, and the time interval after which a restart will be forced (without the user's confirmation). This option is most suitable for workstations where users must be able to select the most convenient time for a restart.

By default, this option is selected.

- **Repeat prompt every (min)**

  If this option is enabled, the application prompts the user to restart the operating system with the specified frequency.

  By default, this option is enabled. The default interval is 5 minutes. Available values are between 1 and 1440 minutes.

  If this option is disabled, the prompt is displayed only once.

- **Restart after (min)**

  After prompting the user, the application forces restart of the operating system upon expiration of the specified time interval.

  By default, this option is enabled. The default delay is 30 minutes. Available values are between 1 and 1440 minutes.

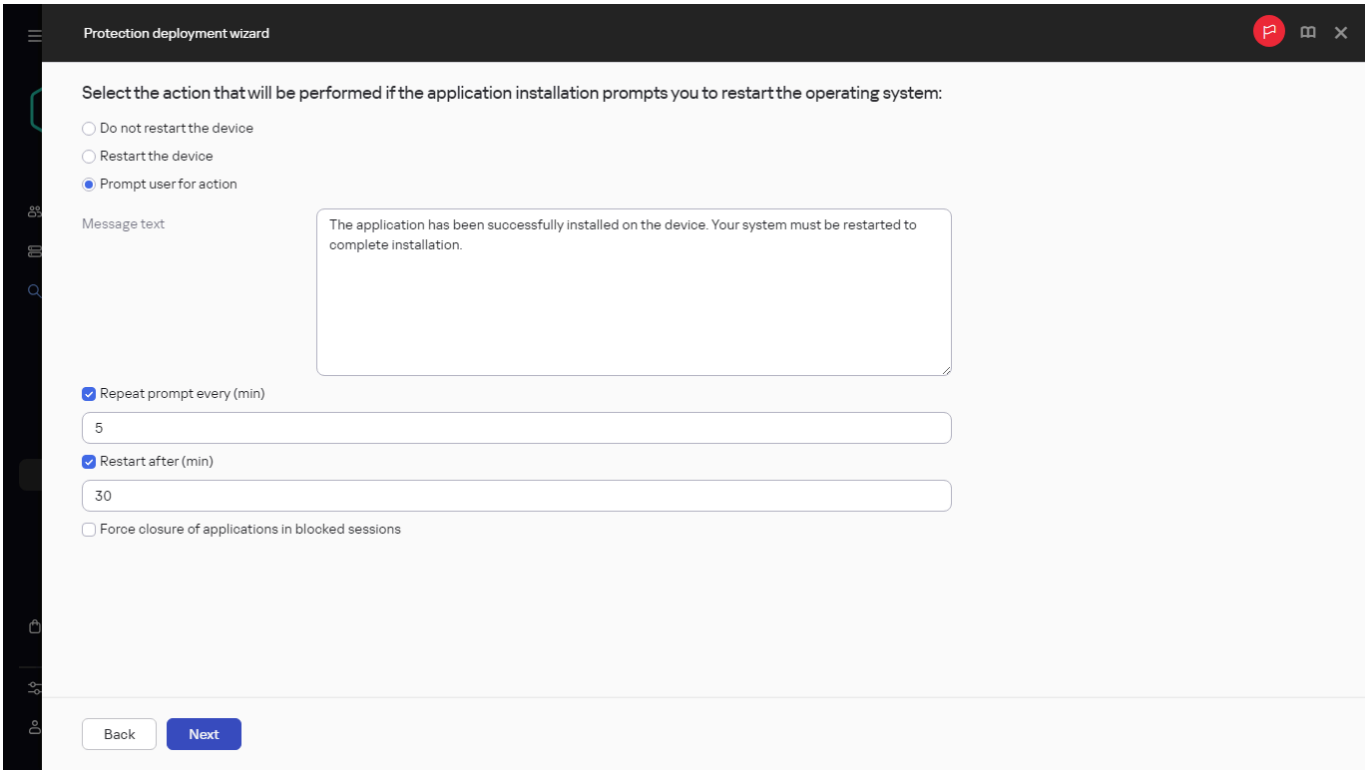
- **Force closure of applications in blocked sessions**

  Running applications may prevent a restart of the client device. For example, if a document is being edited in a word processing application and is not saved, the application does not allow the device to restart.

  If this option is enabled, such applications on a locked device are forced to close before the device restart. As a result, users may lose their unsaved changes.

  If this option is disabled, a locked device is not restarted. The task status on this device states that a device restart is required. Users have to manually close all applications running on locked devices and restart these devices.

  By default, this option is disabled.



Selecting the operating system restart option

Step 8. Removing incompatible applications before installation

This step is only present if the application that you deploy is known to be incompatible with some other applications.

Select the option if you want Open Single Management Platform to automatically remove applications that are incompatible with the application you deploy.

The list of incompatible applications is also displayed.

If you do not select this option, the application will only be installed on devices that have no incompatible applications.

Step 9. Moving devices to Managed devices

Specify whether devices must be moved to an administration group after Network Agent installation.

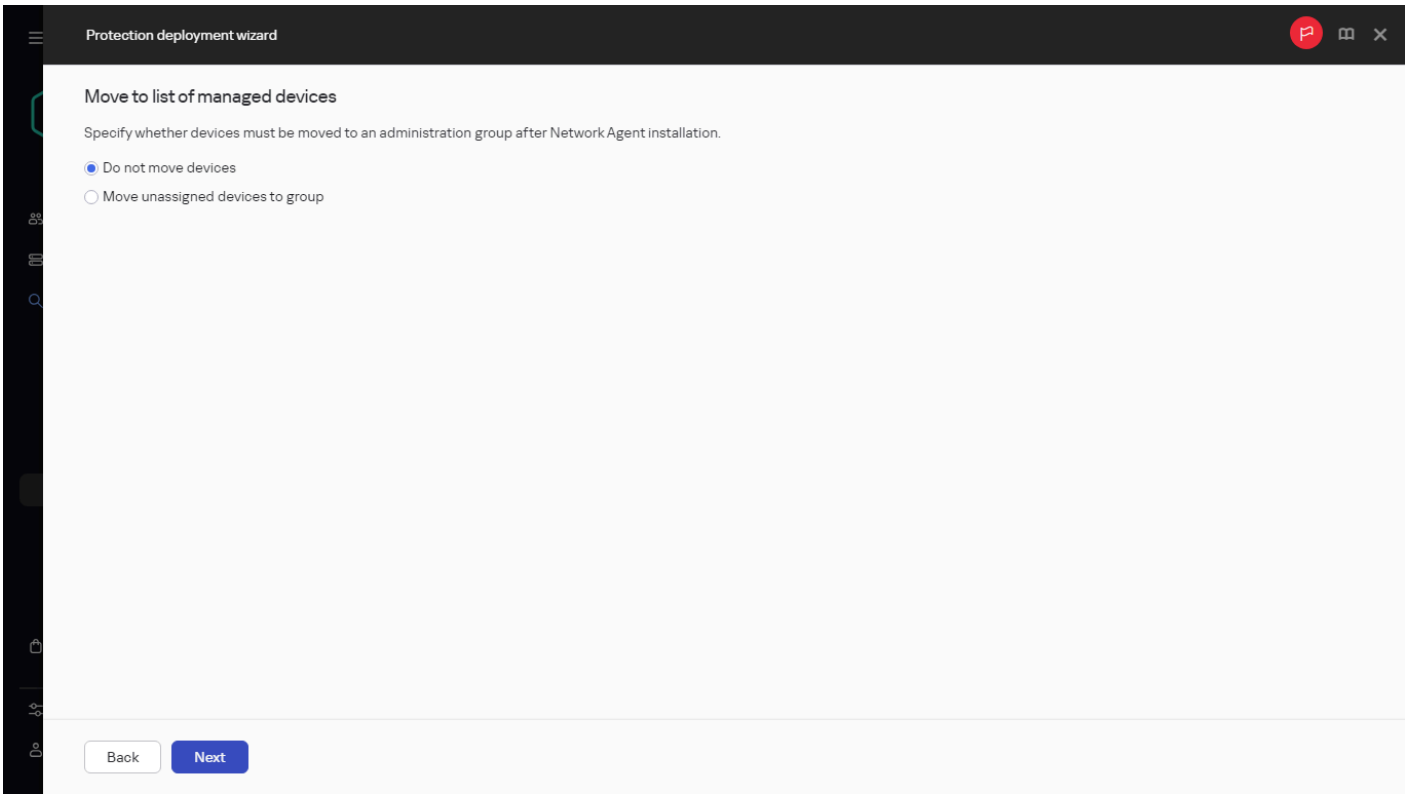
- **Do not move devices**

  The devices remain in the groups in which they are currently located. The devices that have not been placed in any group remain unassigned.

- **Move unassigned devices to group**

  The devices are moved to the administration group that you select.

The **Do not move devices** option is selected by default. For security reasons, you might want to move the devices manually.



Selecting an option to move devices to an administration group

Step 10. Selecting accounts to access devices

If necessary, add the accounts that will be used to start the remote installation task:

- **No account required (Network Agent installed)**

  If this option is selected, you do not have to specify the account under which the application installer will be run. The task will run under the account under which the Administration Server service is running.

  If Network Agent has not been installed on client devices, this option is not available.

- **Account required (Network Agent is not used)**

  Select this option if Network Agent is not installed on the devices for which you assign the remote installation task. In this case, you can specify a user account or an SSH certificate to install the application.

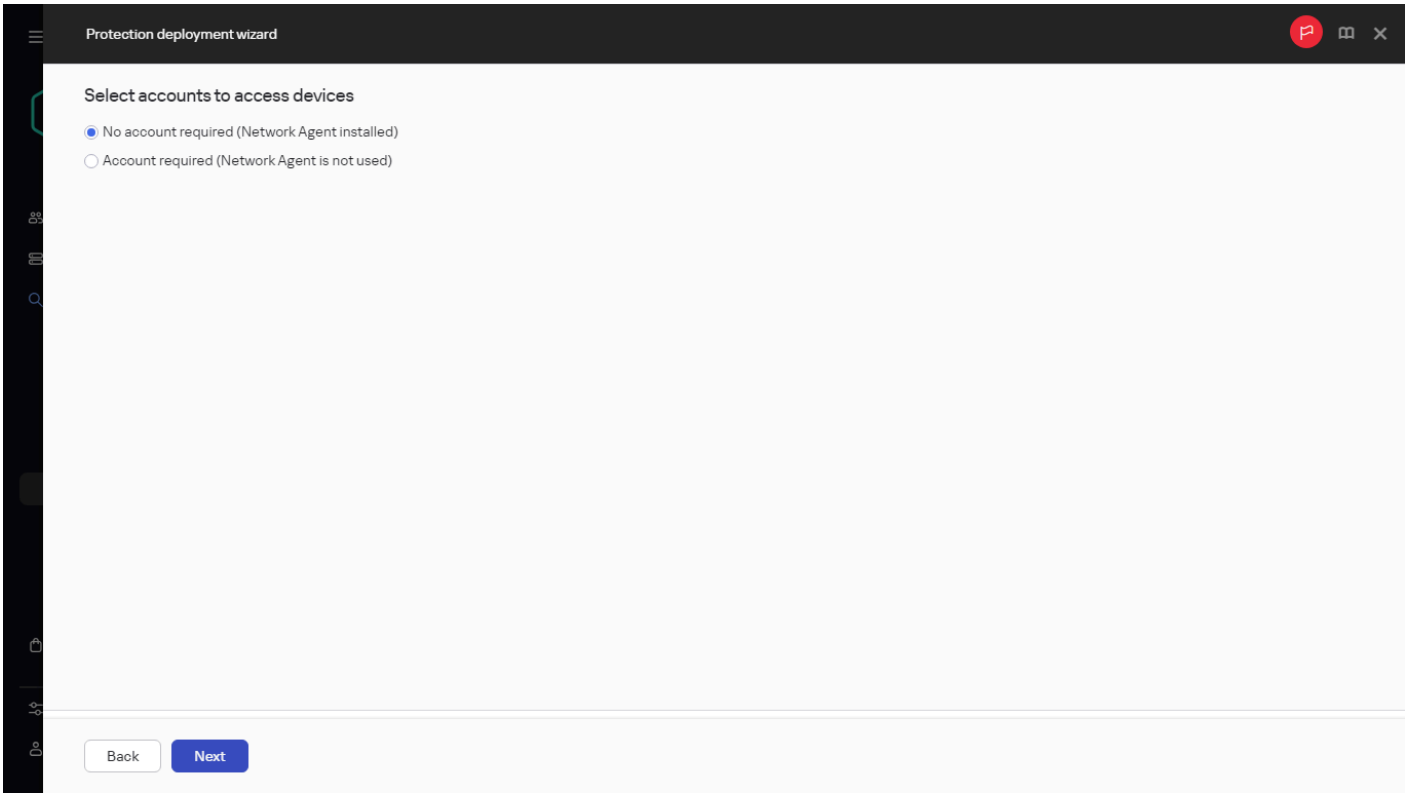
  - **Local Account**. If this option is selected, specify the user account under which the application installer will be run. Click the **Add** button, select **Local Account**, and then specify the user account credentials.

    You can specify multiple user accounts if, for example, none of them have all the required rights on all devices for which you assign the task. In this case, all added accounts are used for running the task, in consecutive order, top-down.

  - **SSH certificate**. If you want to install an application on a Linux-based client device, you can specify an SSH certificate instead of a user account. Click the **Add** button, select **SSH certificate**, and then specify the private and public keys of the certificate.

To generate a private key, you can use the ssh-keygen utility. Note that Open Single Management Platform supports the PEM format of private keys, but the ssh-keygen utility generates SSH keys in the OPENSSH format by default. The OPENSSH format is not supported by Open Single Management Platform. To create a private key in the supported PEM format, add the -m PEM option in the ssh-keygen command. For example:

```
ssh-keygen -m PEM -t rsa -b 4096 -C "<user email>"
```



Selecting accounts to access devices

Step 11. Starting installation

This page is the final step of the wizard. At this step, the **Remote installation task** has been successfully created and configured.

By default, the **Run the task after the wizard finishes** option is not selected. If you select this option, the **Remote installation task** will start immediately after you complete the wizard. If you do not select this option, the **Remote installation task** will not start. You can later start this task manually.

Click **OK** to complete the final step of the Protection deployment wizard.

Adding management plug-ins for Kaspersky applications

For remote administration of Kaspersky applications by using OSMP Console, you must use management web plug-ins. Management web plug-ins are installed automatically after you deploy Kaspersky Next XDR Expert.

If you need to reinstall a management web plug-in manually, contact Kaspersky Technical Support to receive the archive with the required management plug-in.

To reinstall a management web plug-in for a Kaspersky application:

1. Move the management web plug-in archive to the administrator host on which the KDT utility is located.

2. If necessary, on the administrator host, export the current version of the configuration file.

   You do not need to export the configuration file if the installation parameters are not added or modified.

3. Run the following command to install the plug-in:

   ```
   ./kdt apply -k <path_to_plugin_archive> -i <path_to_configuration_file>
   ```

   In the command, specify the path to the plug-in archive and the path to the current configuration file. You do not need to specify the path to the configuration file in the command if the installation parameters are not added or modified.

The management web plug-in is reinstalled. Reload OSMP Console to update the plug-in.

You can view the list of components related to OSMP (including management web plug-ins) by using KDT. Also, you can view OSMP Console version and the list of installed management web plug-ins. To do this, in the upper pane of OSMP Console, go to your account menu, and then select **About**.

Removing management web plug-ins

You can remove the management web plug-ins of Kaspersky applications that provide additional functionality for Kaspersky Next XDR Expert. The Kaspersky Next XDR Expert services plug-ins are used for the correct function of Kaspersky Next XDR Expert and cannot be removed (for example, the plug-in of Incident Response Platform).

To remove a management web plug-in:

1. If needed, run the following command to obtain the name of the plug-in that you want to remove:

   ```
   ./kdt state
   ```

   The list of components is displayed.

2. On the administrator host, run the following command. Specify the name of the plug-in that you want to remove:

   ```
   ./kdt remove --cnab <plug-in_name>
   ```

The specified management web plug-in is removed by KDT.

Viewing the list of components integrated in Open Single Management Platform

You can view the list of components integrated in OSMP (including management web plug-ins) by using KDT.

To view the list of components,

On the administrator host on which KDT is located, run the following command:

```
./kdt state
```

The list of components integrated in OSMP (including management web plug-ins) is displayed in the command line window.

Viewing names, parameters, and custom actions of Kaspersky Next XDR Expert components

KDT allows you to view the list of the Kaspersky Next XDR Expert components that are contained in the transport archive, as well as the list of installed components. Also, you can view the parameter list and the custom action list of a Kaspersky Next XDR Expert component. If custom actions are available for the component, you can also view the description and parameters of the specified custom action by using KDT.

Custom action is an action that allows you to perform additional operations specific to the Kaspersky Next XDR Expert component (except installation, update, deletion). For example, recovering Administration Server data and increasing the amount of disk space used for Administration Server and its logs are performed by using custom actions.

A custom action is run by using KDT as follows:

```
./kdt invoke <component_name> --action <custom_action> --param
<custom_action_parameter>
```

To view the list of Kaspersky Next XDR Expert components included in the transport archive,

On the administrator host where the KDT utility is located, run the following command. In the command, specify the path to the transport archive and its name:

```
./kdt describe -k <transport_archive_name_with_path>
```

To view the list of Kaspersky Next XDR Expert components,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt describe
```

The lists of Kaspersky Next XDR Expert components are displayed.

To view the parameter list and the custom action list of the Kaspersky Next XDR Expert component,

On the administrator host where the KDT utility is located, run the following command and specify the name of the Kaspersky Next XDR Expert component:

```
./kdt describe <component_name>
```

The lists of the parameters and custom actions available for the specified component are displayed.

To view the description and the parameter list of the custom action,

On the administrator host where the KDT utility is located, run the following command and specify the Kaspersky Next XDR Expert component name and its command:

```
./kdt describe <component_name> <custom_action>
```

The description and the parameter list of the specified component custom action are displayed.

Downloading and creating installation packages for Kaspersky applications

You can create installation packages for Kaspersky applications from Kaspersky web servers if your Administration Server has access to the internet.

To download and create installation package for Kaspersky application:

1. Do one of the following:

   - In the main menu, go to **Discovery & deployment** → **Deployment & assignment** → **Installation packages**.
   - In the main menu, go to **Operations** → **Repositories** → **Installation packages**.

   You can also view notifications about new packages for Kaspersky applications in the list of onscreen notifications. If there are notifications about a new package, you can click the link next to the notification and proceed to the list of available installation packages.

   A list of installation packages available on Administration Server is displayed.

2. Click **Add**.

   The New package wizard starts. Proceed through the wizard by using the **Next** button.

3. Select **Create an installation package for a Kaspersky application**.

   A list of available installation packages on Kaspersky web servers appears. The list contains installation packages only for those applications that are compatible with the current version of Open Single Management Platform. By default, the list only shows applications with the language that matches the interface language. To display other languages, click **Filter** and configure the filter settings.

4. Click the name of an installation package, for example, Kaspersky Endpoint Security for Linux.

   A window opens with information about the installation package.

   > You can download and use an installation package which includes cryptographic tools that implement strong encryption, if it complies with applicable laws and regulations. To download the installation package of Kaspersky Endpoint Security for Windows valid for the needs of your organization, consult the legislation of the country where the client devices of your organization are located.

5. Read the information and click the **Download and create installation package** button.

   If a distribution package can not be converted to an installation package, the **Download distribution package** button instead of the **Download and create installation package** is displayed.

The downloading of the installation package to Administration Server starts. You can close the wizard's window. In this case, the download process will continue in background mode. You can track the installation package download status as well as filter and sort the statuses in the **Package status** column.

If the download process stops and the download status switches to **Accept EULA**, then click the installation package name, and then proceed to step 6 of the instruction. By default, the installation packages with **In progress** and **Accept EULA** statuses are placed in the beginning of the list.

> If the size of data contained in the selected distribution package exceeds the current limit, an error message is displayed. You can change the limit value and then proceed with the installation package creation.

6. For some Kaspersky applications, during the download process the **Show EULA** button is displayed. If it is displayed, do the following:

   a. Click the **Show EULA** button to read the End User License Agreement (EULA).

   b. Read the EULA that is displayed on the screen, and click **Accept**.

      The downloading continues after you accept the EULA. If you click **Decline**, the download is stopped.

7. When the downloading is complete, click the **Close** button.

The installation package is displayed in the list of installation packages.

Creating installation packages from a file

You can use custom installation packages to do the following:

- To install any application (such as a text editor) on a client device, for example, by means of a task.
- To create a stand-alone installation package.

A custom installation package is a folder with a set of files. The source to create a custom installation package is an *archive file*. The archive file contains a file or files that must be included in the custom installation package.

While creating a custom installation package, you can specify command-line parameters, for example, to install the application in silent mode.

To create a custom installation package:

1. Do one of the following:

   - In the main menu, go to **Discovery & deployment** → **Deployment & assignment** → **Installation packages**.
   - In the main menu, go to **Operations** → **Repositories** → **Installation packages**.

   A list of installation packages available on the Administration Server is displayed.

2. Click **Add**.

   The New package wizard starts. Proceed through the wizard by using the **Next** button.

3. Select **Create an installation package from a file**.

4. Specify the package name and click the **Browse** button.

5. In the window that opens, choose an archive file located on the available disks.

   You can upload a ZIP, CAB, TAR, or TAR.GZ archive file. It is not possible to create an installation package from an SFX (self-extracting archive) file.

   File upload to the Administration Server starts.

6. If you specified a file of a Kaspersky application, you may be prompted to read and accept the End User License Agreement (EULA) for the application. To continue, you must accept the EULA. Select the **Accept the terms and conditions of this End User License Agreement** option only if you have fully read, understand and accept the terms of the EULA.

   Additionally, you may be prompted to read and accept the Privacy Policy. To continue, you must accept the Privacy Policy. Select the **I accept the Privacy Policy** option only if you understand and agree that your data will be handled and transmitted (including to third countries) as described in the Privacy Policy.

7. Select a file (from the list of files that are extracted from the chosen archive file) and specify the command-line parameters of an executable file.

   You can specify command-line parameters to install the application from the installation package in a silent mode. Specifying command-line parameters is optional.

   The process to create the installation package is started.

   The wizard informs you when the process is finished.

   If the installation package is not created, an appropriate message is displayed.

8. Click the **Finish** button to close the wizard.

The installation package appears in the list of installation packages.

In the list of installation packages available on Administration Server, by clicking the link with the name of a custom installation package, you can:

- View the following properties of an installation package:
  - **Name**. Custom installation package name.
  - **Package status**. Installation package download status.
  - **Source**. Application vendor name.
  - **Application**. Application name packed into the custom installation package.
  - **Version**. Application version.
  - **Language**. Language of the application packed into the custom installation package.
  - **Size (MB)**. Size of the installation package.
  - **OS**. Type of the operating system for which the installation package is intended.
  - **Created**. Installation package creation date.

- **Modified**. Installation package modification date.
- **Type**. Type of the installation package.

- Change the command-line parameters.

Creating stand-alone installation packages

You and device users in your organization can use stand-alone installation packages to install applications on devices manually.

A stand-alone installation package is an executable file that you can store on the Web Server or in the shared folder, send by email, or transfer to a client device by another method. On the client device, the user can run the received file locally to install an application without involving Open Single Management Platform. You can create stand-alone installation packages for Kaspersky applications and for third-party applications. To create a stand-alone installation package for a third-party application you must create a custom installation package.

If the Administration Server has access to Internet, the **Download updates to the Administration Server repository** task is used to download dependencies required for creating Network Agent stand-alone installation packages. If the Administration Server does not have access to Internet, perform the following steps:

1. Execute the following command to create the RIUtilsArchiveDirUsed flag:

   ```
   /opt/kaspersky/ksc64/sbin/klscflag -fset -pv klserver -n RIUtilsArchiveDirUsed -t d -v 1
   ```

2. Execute the following command to restart the Administration Server service:

   ```
   systemctl restart kladminserver_srv.service
   ```

3. Download the following archives from the Kaspersky website and place the archives in the /var/opt/kaspersky/klnagent_srv/1093/.working/ri_utils_archives folder:

   - ri_utils_win32.zip
   - ri_utils_arm64.zip

   Do not rename the archives.

The Administration Server now has the required dependencies.

Be sure that stand-alone installation package is not available for third persons.

To create a stand-alone installation package:

1. Do one of the following:

   - In the main menu, go to **Discovery & deployment** → **Deployment & assignment** → **Installation packages**.
   - In the main menu, go to **Operations** → **Repositories** → **Installation packages**.

   A list of installation packages available on Administration Server is displayed.

2. In the list of installation packages, select an installation package and, above the list, click the **Deploy** button.

3. Select the **Using a stand-alone package** option.

   The Stand-alone installation package creation wizard starts. Proceed through the wizard by using the **Next** button.

4. Make sure that the **Install Network Agent together with this application** option is enabled if you want to install Network Agent together with the selected application.

   > The option is displayed depending on the application for which you create a stand-alone installation package.

   By default, this option is enabled. It is recommended to enable this option if you are not sure whether Network Agent is installed on the device. If Network Agent is already installed on the device, after the stand-alone installation package with Network Agent installed Network Agent will be updated to the newer version.

   If you disable this option, Network Agent will not be installed on the device and the device will be unmanaged.

   If a stand-alone installation package for the selected application already exists on Administration Server, the wizard informs you about this fact. In this case, you must select one of the following actions:

   - **Create stand-alone installation package**. Select this option, for example, if you want to create a stand-alone installation package for a new application version and also want to retain a stand-alone installation package that you created for a previous application version. The new stand-alone installation package is placed in another folder.

   - **Use existing stand-alone installation package**. Select this option if you want to use an existing stand-alone installation package. The process of package creation will not be started.

   - **Rebuild existing stand-alone installation package**. Select this option if you want to create a stand-alone installation package for the same application again. The stand-alone installation package is placed in the same folder.

5. On the **Move to list of managed devices** step, the **Do not move devices** option is selected by default. If you do not want to move the client device to any administration group after Network Agent installation, do not change choice of option.

   If you want to move client device after Network Agent installation, select the **Move unassigned devices to this group** option and specify an administration group to which you want to move the client device. By default, the device is moved to the **Managed devices** group.

6. When the process of the stand-alone installation package creation is finished, click the **FINISH** button.

   The Stand-alone Installation Package Creation Wizard closes.

The stand-alone installation package is created and placed on the Web Server. You can view the list of stand-alone packages by clicking the **View the list of stand-alone packages** button above the list of installation packages.

Changing the limit on the size of custom installation package data

The total size of data unpacked during creation of a custom installation package is limited. The default limit is 1 GB.

If you attempt to upload an archive file that contains data exceeding the current limit, an error message is displayed. You might have to increase this limit value when creating installation packages from large distribution packages.

To change the limit value for the custom installation package size,

On the administrator host where the KDT utility is located, run the following command:

```
./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv klserver -n
MaxArchivePkgSize -t d -v <number of bytes>'
```

Where <number of bytes> is a number of bytes in hexadecimal or decimal format.

For example, if the required limit is 2 GB, you can specify the decimal value 2147483648 or the hexadecimal value 0x80000000. In this case, for a local installation of Administration Server, you can use the following command:

```
./kdt invoke ksc --action klscflag --param klscflag_param=' -fset -pv klserver -n
MaxArchivePkgSize -t d -v 2147483648'
```

The limit on the size of custom installation package data is changed.

Installing Network Agent for Linux in silent mode (with an answer file)

You can install Network Agent on Linux devices by using an answer file—a text file that contains a custom set of installation parameters: variables and their respective values. Using this answer file allows you to run an installation in silent mode, that is, without user participation.

Before installing Network Agent on a Linux-based device, you must complete the general preparatory steps that are required to all Linux-based devices.

After completing the general steps, you can follow the additional instructions specific to your operating system:

- Preparing a device running SUSE Linux Enterprise Server 15
- Preparing a device running Astra Linux in the closed software environment mode
- Preparing a device running Ubuntu Server/Desktop version 10.04 and 16.04
- Preparing a device running a Linux operating system with glibc 2.5

To perform installation of Network Agent for Linux in silent mode:

1. Read the End User License Agreement. Follow the steps below only if you understand and accept the terms of the End User License Agreement.
2. Create the answer file (in TXT format). Add to the answer file a list of variables in the VARIABLE_NAME=variable_value format, each variable on a separate line.

   For correct usage of the answer file, you must include in it a minimum set of the three required variables:

   - KLNAGENT_SERVER
   - KLNAGENT_AUTOINSTALL
   - EULA_ACCEPTED

You can also add any optional variables to use more specific parameters of your remote installation. The following table lists all of the variables that can be included in the answer file:

Variables of the answer file used as parameters of Network Agent for Linux installation in silent mode

Variables of the answer file used as parameters of Network Agent for Linux installation in silent mode

Variable name	Required	Description	Possible values
KLNAGENT_SERVER	Yes	Contains the Administration Server name presented as fully qualified domain name (FQDN) or IP address.	DNS name or IP address.
KLNAGENT_AUTOINSTALL	Yes	Defines whether silent installation mode is enabled.	1—Silent mode is enabled; the user is not prompted for any actions during installation. Other—Silent mode is disabled; the user may be prompted for actions during installation.
EULA_ACCEPTED	Yes	Defines whether the user accepts the End User License Agreement (EULA) of Network Agent; when missing, can be interpreted as non-acceptance of the EULA.	1—I confirm that I have fully read, understand, and accept the terms and conditions of this End User License Agreement. Other or not specified—I do not accept the terms of the License Agreement (installation is not performed).
KLNAGENT_PROXY_USE	No	Defines whether connection with the Administration Server will use proxy settings. The default value is 0.	1—Proxy settings are used. Other—Proxy settings are not used.
KLNAGENT_PROXY_ADDR	No	Defines the address of the proxy server used for connection with the Administration Server.	DNS name or IP address.
KLNAGENT_PROXY_LOGIN	No	Defines the user name used for login to the proxy server.	Any existing user name.
KLNAGENT_PROXY_PASSWORD	No	Defines the user password used for login to the proxy server.	Any set of alphanumeric characters allowed by the password format in the operating system.
KLNAGENT_VM_VDI	No	Defines whether Network Agent is installed on an image for creation of dynamic virtual machines.	1—Network Agent is installed on an image, which is subsequently used for creation of dynamic virtual machines. Other—No image is used during installation.
KLNAGENT_VM_OPTIMIZE	No	Defines whether the Network Agent settings are optimal for hypervisor.	1—The default local settings of Network Agent are modified so that they allow optimized usage on hypervisor.
KLNAGENT_TAGS	No	Lists the tags assigned to the Network Agent instance.	One or multiple tag names separated with semicolon.
KLNAGENT_UDP_PORT	No	Defines the UDP port used by Network Agent. The default value is 15000.	Any existing port number.
KLNAGENT_PORT	No	Defines the non-TLS port used by Network Agent. The default value is 14000.	Any existing port number.
KLNAGENT_SSLPORT	No	Defines the TLS port used by Network Agent. The default value is 13000.	Any existing port number.

KLNAGENT_USESSL	No	Defines whether Transport Layer Security (TLS) is used for connection.	1 (default)—TLS is used. Other—TLS is not used.
KLNAGENT_GW_MODE	No	Defines whether connection gateway is used.	1 (default)—The current settings are not modified (at the first call, no connection gateway is specified). 2—No connection gateway is used. 3—Connection gateway is used. KLNAGENT_GW_ADDRESS is required. 4—The Network Agent instance is used as connection gateway in demilitarized zone (DMZ). Server certificate is required.
KLNAGENT_GW_ADDRESS	No	Defines the address of the connection gateway. The value is applicable only if KLNAGENT_GW_MODE=3.	DNS name or IP address.
KLNAGENT_DEVICEOWNER_REGISTRATION_START	No	Allows to run the user registration as a device owner utility after Network Agent installation. If turned off, then the registration as a device owner is not available to a user.	1—The user registration as a device owner utility will run after Network Agent installation. Other—Turned off.
PTCH_ALLOW_APPLY_NONAPROVED_PATCHES	No	Defines whether to install the downloaded updates automatically for Network Agent with the *Undefined* status.	true (default)—Updates are installed automatically. false—Updates are not installed automatically.

3. Run the Network Agent installation. The KLAUTOANSWERS environment variable, which specifies the path to the answer file, is passed to the installation command. Run one of the following command:

   - To install Network Agent from an RPM package to a 32-bit operating system, execute the following command:
     `sudo KLAUTOANSWERS=<path to answer file> yum install ./klnagent-<build number>.i386.rpm`

   - To install Network Agent from an RPM package to a 64-bit operating system, execute the following command:
     `sudo KLAUTOANSWERS=<path to answer file> yum install ./klnagent64-<build number>.x86_64.rpm`

   - To install Network Agent from an RPM package on a 64-bit operating system for the Arm architecture, execute the following command:
     `sudo KLAUTOANSWERS=<path to answer file> yum install ./klnagent64-<build number>.aarch64.rpm`

   - To install Network Agent from a DEB package to a 32-bit operating system, execute the following command:
     `sudo KLAUTOANSWERS=<path to answer file> apt install ./klnagent_<build number>_i386.deb`

- To install Network Agent from a DEB package to a 64-bit operating system, execute the following command:
  `sudo KLAUTOANSWERS=< path to answer file > apt install ./klnagent64_< build number >_amd64.deb`

- To install Network Agent from a DEB package on a 64-bit operating system for the Arm architecture, execute the following command:
  `sudo KLAUTOANSWERS=< path to answer file > apt install ./klnagent64_< build number >_arm64.deb`

Depending on the operating system configuration, you can also use the rpm, dpkg, or dnf package managers to install Network Agent.

Installation of Network Agent for Linux starts in silent mode; the user is not prompted for any actions during the process.

Preparing a device running Astra Linux in the closed software environment mode for installation of Network Agent

Prior to the installation of Network Agent on a device running Astra Linux in the closed software environment mode, you must perform two preparation procedures—the one in the instructions below and general preparation steps for any Linux device.

Run the commands provided in this instruction under an account with root privileges.

To prepare a device running Astra Linux in the closed software environment mode for installation of Network Agent:

1. Open the `/etc/digsig/digsig_initramfs.conf` file, and then specify the following setting:

   ```
   DIGSIG_ELF_MODE=1
   ```

2. In the command line, run the following command to install the compatibility package:

   ```
   apt install astra-digsig-oldkeys
   ```

3. Create a directory for the application key:

   ```
   mkdir -p /etc/digsig/keys/legacy/kaspersky/
   ```

4. Place the application key /opt/kaspersky/klnagent64/share/kaspersky_astra_pub_key.gpg in the directory created in the previous step:

   ```
   cp kaspersky_astra_pub_key.gpg /etc/digsig/keys/legacy/kaspersky/
   ```

   If the Open Single Management Platform distribution kit does not include the kaspersky_astra_pub_key.gpg application key, you can download it by clicking the link: https://media.kaspersky.com/utilities/CorporateUtilities/kaspersky_astra_pub_key.gpg.

5. Update the RAM disks:

   ```
   update-initramfs -u -k all
   ```

   Reboot the system.

The Linux device is ready for installation of Network Agent. You can install Network Agent remotely, in interactive mode, or in silent mode.

Viewing the list of stand-alone installation packages

You can view the list of stand-alone installation packages and properties of each stand-alone installation package.

To view the list of stand-alone installation packages for all installation packages:

Above the list, click the **View the list of stand-alone packages** button.

In the list of stand-alone installation packages, their properties are displayed as follows:

- **Package name**. Stand-alone installation package name that is automatically formed as the application name included in the package and the application version.
- **Application name**. Application name included in the stand-alone installation package.
- **Application version**.
- **Network Agent installation package name**. The property is displayed only if Network Agent is included in the stand-alone installation package.
- **Network Agent version**. The property is displayed only if Network Agent is included in the stand-alone installation package.
- **Size**. File size in MB.
- **Group**. Name of the group to which the client device is moved after Network Agent installation.
- **Created**. Date and time of the stand-alone installation package creation.
- **Modified**. Date and time of the stand-alone installation package modification.
- **Path**. Full path to the folder where the stand-alone installation package is located.
- **Web address**. Web address of the stand-alone installation package location.
- **File hash**. The property is used to certify that the stand-alone installation package was not changed by third-party persons and a user has the same file you have created and transferred to the user.

Information about the size, file hash, and path is not available in the list of installation packages for the Network Agent stand-alone package or for any packages created on virtual Administration Servers. These stand-alone packages are generated at the time of download.

To view the list of stand-alone installation packages for specific installation package:

Select the installation package in the list and, above the list, click the **View the list of stand-alone packages** button.

In the list of stand-alone installation packages, you can do the following:

- Publish a stand-alone installation package on the Web Server by clicking the **Publish** button. Published stand-alone installation package is available for downloading for users whom you sent the link to the stand-alone installation package.
- Cancel publication of a stand-alone installation package on the Web Server by clicking the **Unpublish** button. Unpublished stand-alone installation package is available for downloading only for you and other administrators.
- Download a stand-alone installation package to your device by clicking the **Download** button.
- Send email with the link to a stand-alone installation package by clicking the **Send by email** button.
- Remove a stand-alone installation package by clicking the **Remove** button.

Distributing installation packages to secondary Administration Servers

Open Single Management Platform allows you to create installation packages for Kaspersky applications and for third-party applications, as well as distribute installation packages to client devices and install applications from the packages. To optimize the load on the primary Administration Server, you can distribute installation packages to secondary Administration Servers. After that, the secondary Servers transmit the packages to client devices, and then you can perform the remote installation of the applications on your client devices.

To distribute installation packages to secondary Administration Servers:

1. Make sure that the secondary Administration Servers are connected to the primary Administration Server.
2. In the main menu, go to **Asset management** → **Tasks**.

   The list of tasks is displayed.
3. Click the **Add** button.

   The New task wizard starts. Follow the steps of the wizard.
4. On the **New task settings** page, from the **Application** drop-down list, select **Kaspersky Security Center**. Then, from the **Task type** drop-down list, select **Distribute installation package**, and then specify the task name.
5. On the **Task scope** page, select the devices to which the task is assigned in one of the following ways:
   - If you want to create a task for all secondary Administration Servers in a specific administration group, select this group, and then create a group task for it.
   - If you want to create a task for specific secondary Administration Servers, select these Servers, and then create a task for them.
6. On the **Distributed installation packages** page, select the installation packages that are to be copied to the secondary Administration Servers.
7. Specify an account to run the *Distribute installation package* task under this account. You can use your account and keep the **Default account** option enabled. Alternatively, you can specify that the task should be run under another account that has the necessary access rights. To do this, select the **Specify account** option, and then enter the credentials of that account.

8. On the **Finish task creation** page, you can enable the **Open task details when creation is complete** option to open the task properties window, and then modify the default task settings. Otherwise, you can configure the task settings later, at any time.

9. Click the **Finish** button.

   The task created for distributing installation packages to the secondary Administration Servers is displayed in the task list.

10. You can run the task manually or wait for it to launch according to the schedule that you specified in the task settings.

After the task is completed, the selected installation packages are copied to the specified secondary Administration Servers.

Preparing a Linux device for installation of Network Agent

Prior to the installation of Network Agent on a Linux device, you must perform preparation steps that are required for any Linux device.

To prepare a device running Linux for remote installation of Network Agent:

1. Make sure that the device on which you want to install Network Agent for Linux is running one of the supported Linux distributions.

2. If you want to install Network Agent on AltOS in the closed software environment mode, disable the closed software environment mode until the installation is finished.

3. Download the Network Agent installation package by using the application interface or from the Kaspersky website.

   If you need to install Kaspersky Industrial CyberSecurity for Linux Nodes on a device running a Linux operating system with glibc 2.5, you must download one of the following installation packages from the Kaspersky website:

   - klnagent64_16.1.0-6517_amd64_glibc2.5.deb
   - klnagent64-16.1.0-6517.x86_64_glibc2.5.rpm
   - klnagent_16.1.0-6517_i386_glibc2.5.deb
   - klnagent-16.1.0-6517.i386_glibc2.5.rpm

   These installation packages are available for the following operating systems with glibc 2.5:

   - CentOS 5.2 32-bit/64-bit
   - CentOS 5.4 32-bit/64-bit
   - CentOS 6.4 32-bit/64-bit
   - CentOS 6.6 32-bit/64-bit
   - Astra Linux Special Edition RUSB.10015-01 (operational update 1.4) 64-bit

   You must have the Kaspersky Industrial CyberSecurity for Linux Nodes license to get the installation package.

4. Make sure that the following software is installed on the target Linux device:

   - Sudo (for Ubuntu 10.04, Sudo version is 1.7.2p1 or later)
   - Perl language interpreter version 5.10 or later
   - ps utility

5. Test the device configuration:

   a. Check whether you can connect to the device through an SSH client (such as PuTTY).

      If you cannot connect to the device, open the `/etc/ssh/sshd_config` file and make sure that the following settings have the respective values listed below:

      ```
      PasswordAuthentication no
      ChallengeResponseAuthentication yes
      ```

      > Do not modify the /etc/ssh/sshd_config file if you can connect to the device with no issues; otherwise, you may encounter SSH authentication failure when running a remote installation task.

      Save the file (if necessary) and restart the SSH service by using the `sudo service ssh restart` command.

   b. Ensure that port TCP 22 is open and accessible on a Linux-based unassigned device.

      TCP port 22 must be available for successful installation of Network Agent for Linux.

   c. Disable the sudo password for the user account under which the device is to be connected.

   d. Use the `visudo` command in sudo to open the sudoers configuration file.

      In the file you have opened, add the following line to the end of the file: `<username> ALL = (ALL) NOPASSWD: ALL`. In this case, `<username>` is the user account which is to be used for the device connection using SSH. If you are using the Astra Linux operating system, in the /etc/sudoers file, add the last line with the following text: `%astra-admin ALL=(ALL:ALL) NOPASSWD: ALL`

   e. Save the sudoers file and then close it.

   f. Connect to the device again through SSH and make sure that the Sudo service does not prompt you to enter a password; you can do this using the `sudo whoami` command.

6. If you want to install Network Agent remotely on devices running operating system with the systemd initialization system, open the `/etc/systemd/logind.conf` file, and then do one of the following:

   - Specify `no` as a value for the `KillUserProcesses` setting: `KillUserProcesses=no`.
   - For the `KillExcludeUsers` setting, type the user name of the account under which the remote installation is to be performed, for example, `KillExcludeUsers=root`.

   Astra Linux target device

   > If the target device is running Astra Linux, add `export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin` string in the `/home/<username>/.bashrc` file, where `<username>` is the user account which is to be used for the device connection using SSH.

OSnova target device

If the target device is running OSnova, do the following:

a. Open the `/usr/lib/systemd/logind.conf/10-enable-kill-user-processes.conf` file, and then comment out the `#KillUserProcess=yes` line.

b. Open the `/usr/lib/NESS/pam-user-session` file, and then comment out the `#loginctl terminate-session "$XDG_SESSION_ID"` line.

To apply the changed setting, restart the Linux device or execute the following command:

```
sudo systemctl restart systemd-logind.service
```

7. If you want to install Network Agent on devices with the SUSE Linux Enterprise Server 15 operating system, install the insserv-compat package first to configure Network Agent.

8. If you want to install Network Agent on devices that have the Astra Linux operating system running in the closed software environment mode, perform additional steps to prepare Astra Linux devices.

9. If you want to install Network Agent on devices running Ubuntu Server or Ubuntu Desktop version 10.04, perform additional steps to prepare these devices.

10. If you want to install Kaspersky Industrial CyberSecurity for Linux Nodes on a device running a Linux operating system with glibc 2.5, perform additional steps to prepare the target devices.

11. If you're installing Network Agent on AltOS in the closed software environment mode, re-enable the closed software environment mode.

12. If you want to install Network Agent on devices that use the operating system RED OS 7.3.4 or later or MSVSPHERE 9.2 or later, install the libxcrypt-compat package for the correct function of Network Agent.

The Linux device is ready for installation of Network Agent. You can install Network Agent remotely, in interactive mode, or in silent mode.

Installing applications using a remote installation task

Open Single Management Platform allows you to install applications on devices remotely, using remote installation tasks. Those tasks are created and assigned to devices through a dedicated wizard. To assign a task more quickly and easily, you can specify devices (up to 1000 devices) in the wizard window in one of the following ways:

- **Assign task to an administration group**. In this case, the task is assigned to devices included in an administration group created earlier.

- **Specify device addresses manually or import addresses from a list**. You can specify DNS names, IP addresses, and IP subnets of devices to which you want to assign the task.

- **Assign task to a device selection**. In this case, the task is assigned to devices included in a selection created earlier. You can specify the default selection or a custom one that you created. You can only select up to 1000 devices.

To avoid issues that may occur during installation of the application on a client device without Network Agent installed, you must proceed as described in forced deployment through the remote installation task of Open Single Management Platform.

Installing an application remotely

This section contains information on how to remotely install an application on devices in an administration group, devices with specific addresses, or a selection of devices.

To install an application on specific devices:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click **Add**.

   The New task wizard starts.
3. In the **Task type** field, select **Install application remotely**.
4. Select one of the following options:

   - **Assign task to an administration group**

     The task is assigned to devices included in an administration group. You can specify one of the existing groups or create a new one.

     For example, you may want to use this option to run a task of sending a message to users if the message is specific for devices included in a specific administration group.

     If a task is assigned to an administration group, the **Security** tab is not displayed in the task properties window because group tasks are subject to the security settings of the groups to which they apply.

   - **Specify device addresses manually or import addresses from a list**

     You can specify DNS names, IP addresses, and IP subnets of devices to which you want to assign the task.

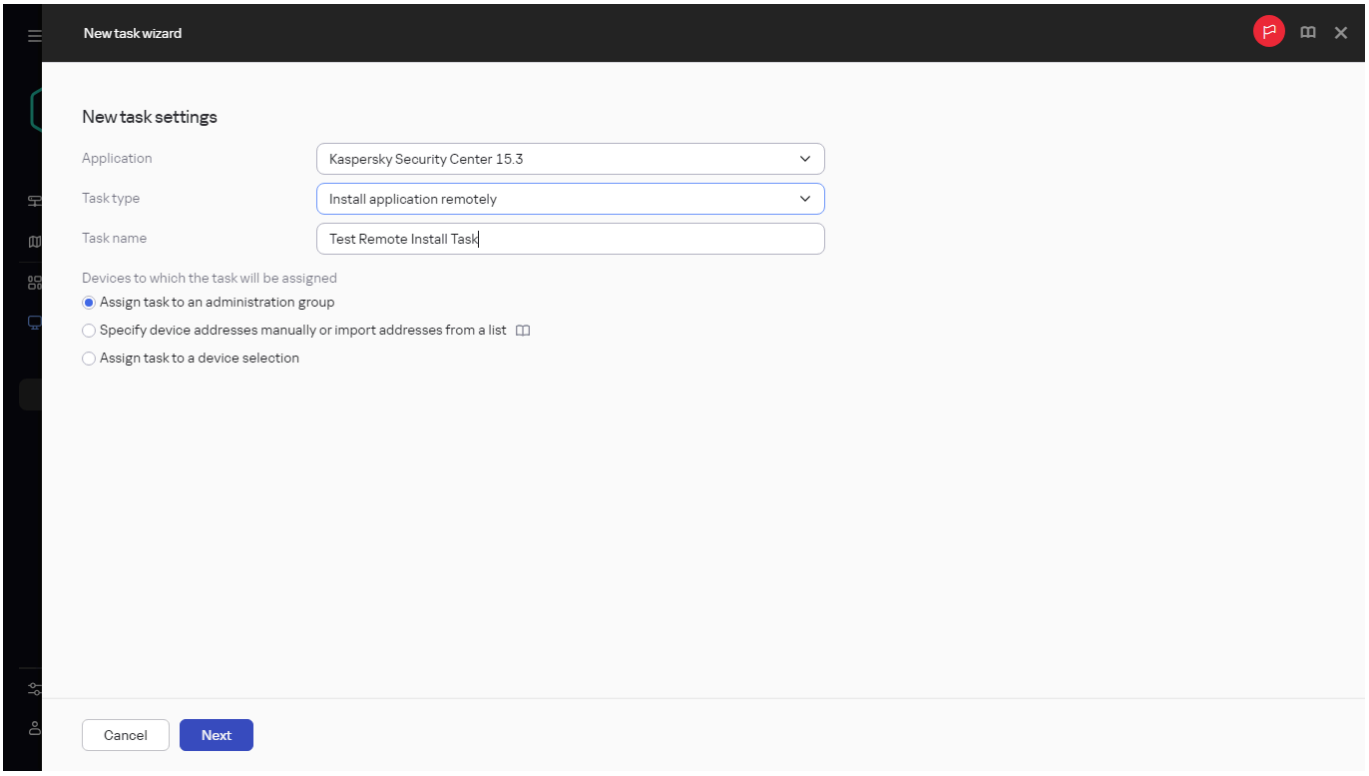
     You may want to use this option to execute a task for a specific subnet. For example, you may want to install a certain application on devices of accountants or to scan devices in a subnet that is probably infected.

   - **Assign task to a device selection**

     The task is assigned to devices included in a device selection. You can specify one of the existing selections.

     For example, you may want to use this option to run a task on devices with a specific operating system version.

The *Install application remotely* task is created for the specified devices. If you selected the **Assign task to an administration group** option, the task is a group one.



Creating the Install application remotely task

5. At the **Task scope** step, specify an administration group, devices with specific addresses, or a device selection.

   The available settings depend on the option selected at the previous step.

6. At the **Installation packages** step, specify the following settings:

   - In the **Select installation package** field, select the installation package of an application that you want to install.

   - In the **Force installation package download** settings group, specify how files that are required for the application installation are distributed to client devices:

     - **Using Network Agent**

       If this option is enabled, installation packages are delivered to client devices by Network Agent installed on those client devices.

       If this option is disabled, installation packages are delivered using the operating system tools of client devices.

       We recommend that you enable this option if the task has been assigned to devices with Network Agents installed.

       By default, this option is enabled.

     - **Using operating system resources through distribution points**

       If this option is enabled, installation packages are transmitted to client devices using operating system tools through distribution points. You can select this option if there is at least one distribution point on the network.

       If the **Using Network Agent** option is enabled, the files are delivered using operating system tools only if Network Agent tools are unavailable.

By default, this option is enabled for remote installation tasks that have been created on a virtual Administration Server.

The only way to install an application for Windows (including Network Agent for Windows) on a device that does not have Network Agent installed is by using a Windows-based distribution point. Therefore, when you install a Windows application:

- Select this option.
- Ensure that a distribution point is assigned for the target client devices.
- Ensure the distribution point is Windows-based.

- **Using operating system resources through Administration Server**

  If this option is enabled, files are transmitted to client devices by using operating system tools of client devices through the Administration Server. You can enable this option if no Network Agent is installed on the client device, but the client device is in the same network as the Administration Server.

  By default, this option is enabled.

- In the **Maximum number of concurrent downloads** field, specify the maximum allowed number of client devices to which Administration Server can simultaneously transmit the files.

- In the **Maximum number of installation attempts** field, specify the maximum allowed number of installer runs.

  If the number of attempts specified in the parameter is exceeded, Open Single Management Platform does not start the installer on the device anymore. To restart the *Install application remotely* task, increase the value of the **Maximum number of installation attempts** parameter and start the task. Alternatively, you can create a new *Install application remotely* task.

- If you migrate from one Kaspersky application to another and your current application is password-protected, enter the password in the **Password to uninstall the current Kaspersky application** field. Note that during the migration, your current Kaspersky application will be uninstalled.

  The **Password to uninstall the current Kaspersky application** field is only available if you have selected the **Using Network Agent** option in the **Force installation package download** settings group.

  You can use the uninstall password only for the Kaspersky Security for Windows Server to Kaspersky Endpoint Security for Windows migration scenario when installing Kaspersky Endpoint Security for Windows by using the Install application remotely task. Using the uninstall password when installing other components may cause installation errors.

  To complete the migration scenario successfully, make sure that the following prerequisites are met:

  - You are using Kaspersky Security Center Network Agent 14.2 for Windows or later.
  - You are installing the application on devices running Windows.

- Define the additional setting:
  - **Do not re-install application if it is already installed**

    If this option is enabled, the selected application will not be re-installed if it has already been installed on this client device.

    If this option is disabled, the application will be installed anyway.

    By default, this option is enabled.

  - **Verify operating system type before downloading**

    Before transmitting the files to client devices, Open Single Management Platform checks if the Installation utility settings are applicable to the operating system of the client device. If the settings are not applicable, Open Single Management Platform does not transmit the files and does not attempt to install the application. For example, to install some application to devices of an administration group that includes devices running various operating systems, you can assign the installation task to the administration group, and then enable this option to skip devices that run an operating system other than the required one.

    When you remotely install Network Agent on unassigned devices, Open Single Management Platform or a distribution point automatically determines the device's operating system type, regardless of the **Verify operating system type before downloading** setting.

  - **Assign package installation in Active Directory group policies**

    If this option is enabled, an installation package is installed by using the Active Directory group policies.

    This option is available if the Network Agent installation package is selected.

    By default, this option is disabled.

  - **Prompt users to close running applications**

    Running applications may prevent a restart of the client device. For example, if a document is being edited in a word processing application and is not saved, the application does not allow the device to restart.

    If this option is enabled, such applications on a locked device are forced to close before the device restart. As a result, users may lose their unsaved changes.

    If this option is disabled, a locked device is not restarted. The task status on this device states that a device restart is required. Users have to manually close all applications running on locked devices and restart these devices.

    By default, this option is disabled.

- Select on which devices you want to install the application:
  - **Install on all devices**

    The application will be installed even on devices managed by other Administration Servers.

    This option is selected by default. You do not have to change this setting if you have only one Administration Server in your network.

- **Install only on devices managed through this Administration Server**

  The application will be installed only on devices managed by this Administration Server. Select this option if you have more than one Administration Server in your network and want to avoid conflicts between them.

- Specify whether devices must be moved to an administration group after installation:

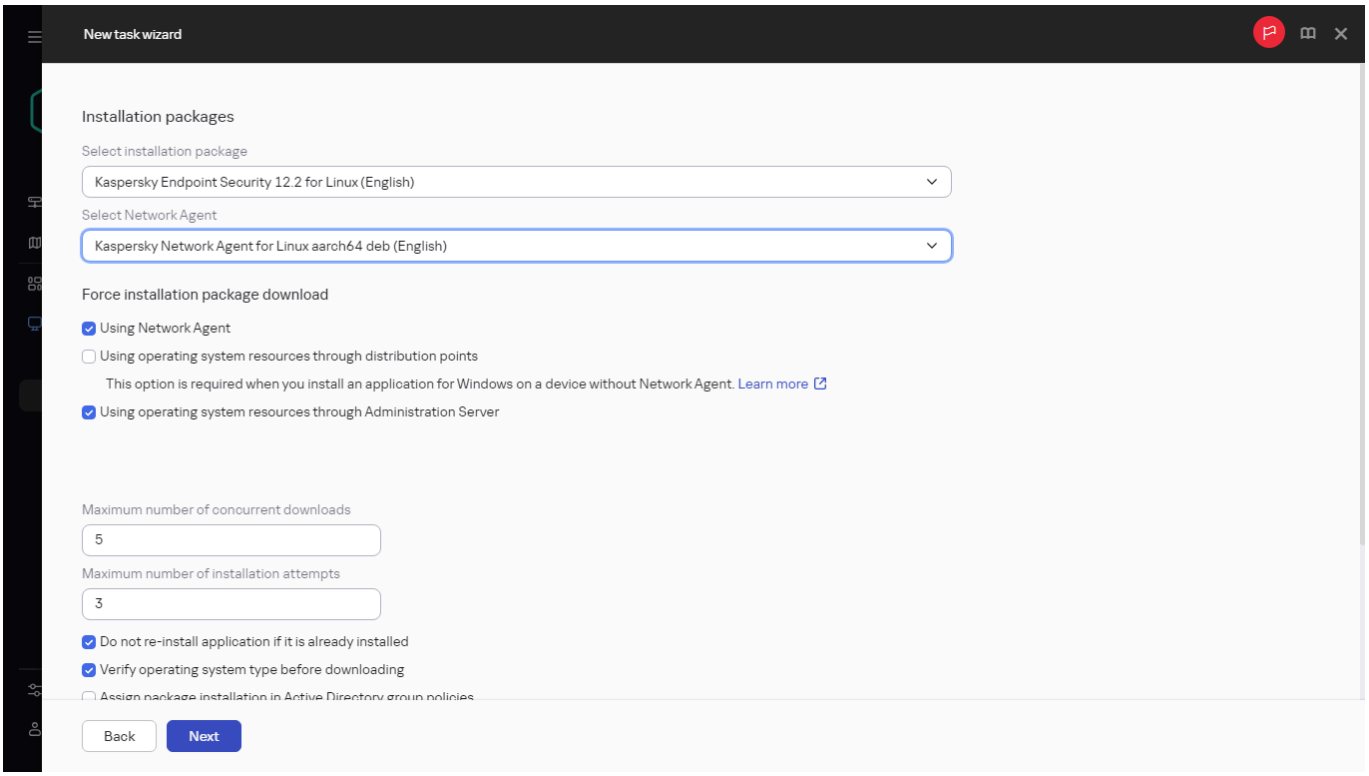
  - **Do not move devices**

    The devices remain in the groups in which they are currently located. The devices that have not been placed in any group remain unassigned.

  - **Move unassigned devices to the selected group (only a single group can be selected)**

    The devices are moved to the administration group that you select.

  Note that the **Do not move devices** option is selected by default. For security reasons, you might want to move the devices manually.



Specifying the task settings

7. At the this step of the wizard, specify whether the devices must be restarted during installation of applications:

  - **Do not restart the device**

    If this option is selected, the device will not be restarted after the security application installation.

- **Restart the device**

  If this option is selected, the device will be restarted after the security application installation.

8. If necessary, at the **Select accounts to access devices** step, add the accounts that will be used to start the *Install application remotely* task:

   - **No account required (Network Agent installed)**

     If this option is selected, you do not have to specify the account under which the application installer will be run. The task will run under the account under which the Administration Server service is running.

     If Network Agent has not been installed on client devices, this option is not available.

   - **Account required (Network Agent is not used)**

     Select this option if Network Agent is not installed on the devices for which you assign the remote installation task. In this case, you can specify a user account or an SSH certificate to install the application.

     - **Local Account**. If this option is selected, specify the user account under which the application installer will be run. Click the **Add** button, select **Local Account**, and then specify the user account credentials.

       You can specify multiple user accounts if, for example, none of them have all the required rights on all devices for which you assign the task. In this case, all added accounts are used for running the task, in consecutive order, top-down.

     - **SSH certificate**. If you want to install an application on a Linux-based client device, you can specify an SSH certificate instead of a user account. Click the **Add** button, select **SSH certificate**, and then specify the private and public keys of the certificate.

       To generate a private key, you can use the ssh-keygen utility. Note that Open Single Management Platform supports the PEM format of private keys, but the ssh-keygen utility generates SSH keys in the OPENSSH format by default. The OPENSSH format is not supported by Open Single Management Platform. To create a private key in the supported PEM format, add the -m PEM option in the ssh-keygen command. For example:

       ```
       ssh-keygen -m PEM -t rsa -b 4096 -C "<user email>"
       ```

9. At the **Finish task creation** step, click the **Finish** button to create the task and close the wizard.

   If you enabled the **Open task details when creation is complete** option, the task settings window opens. In this window, you can check the task parameters, modify them, or configure a task start schedule, if necessary.

10. In the task list, select the task you created, and then click **Start**.

    Alternatively, wait for the task to launch according to the schedule that you specified in the task settings.

When the remote installation task is completed, the selected application is installed on the specified devices.

Installing applications on secondary Administration Servers

If your Administration Server version is 15.4 or later, you can install applications on a secondary Administration Server only if its version is 15.4 or later.

To install an application on secondary Administration Servers:

1. Establish a connection with the Administration Server that controls the relevant secondary Administration Servers.

2. Make sure that the installation package corresponding to the application being installed is available on each of the selected secondary Administration Servers. If you cannot find the installation package on any of the secondary Servers, distribute it. For this purpose, create a task with the **Distribute installation package** task type.

3. Create a task for a remote application installation on secondary Administration Servers. Select the **Install application on secondary Administration Server remotely** task type.

   The New task wizard creates a task for remote installation of the application selected in the wizard on specific secondary Administration Servers.

4. Run the task manually or wait for it to launch according to the schedule that you specified in the task settings.

When the remote installation task is complete, the selected application is installed on the secondary Administration Servers.

Specifying settings for remote installation on Unix devices

When you install an application on a Unix device by using a remote installation task, you can specify Unix-specific settings for the task. These settings are available in the task properties after the task is created.

To specify Unix-specific settings for a remote installation task:

1. In the main menu, go to **Asset management** → **Tasks**.

2. Click the name of the remote installation task for which you want to specify the Unix-specific settings.

   The task properties window opens.

3. Go to **Application settings** → **Unix-specific settings**.

4. Specify the following settings:

   - **Set a password for the root account (only for deployment through SSH)**

     If the `sudo` command cannot be used on the target device without specifying the password, select this option, and then specify the password for the root account. Kaspersky Next XDR Expert transmits the password in an encrypted form to the target device, decrypts the password, and then starts the installation procedure on behalf of the root account with the specified password.

     Kaspersky Next XDR Expert does not use the account or the specified password to create an SSH connection.

   - **Specify the path to a temporary folder with Execute permissions on the target device (only for deployment through SSH)**

     If the /tmp directory on the target device does not have the execute permission, select this option, and then specify the path to the directory with the execute permission. Kaspersky Next XDR Expert uses the specified directory as a temporary directory to access via SSH. The application places the installation package in the directory and runs the installation procedure.

5. Click the **Save** button.

   The specified task settings are saved.

Starting and stopping Kaspersky applications

You can use the *Start or stop application* task for starting and stopping Kaspersky applications on managed devices.

To create the Start or stop application task:

1. In the main menu, go to **Asset management** → **Tasks**.

2. Click **Add**.

   The New task wizard starts. Proceed through the wizard by using the **Next** button.

3. In the **Application** drop-down list, select the application for which you want to create the task.

4. In the **Task type** list, select the **Application activation** task.

5. In the **Task name** field, specify the name of the new task.

   The task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).

6. Select devices to which the task will be assigned.

7. In the **Applications** window, do the following:

   - Select the check boxes next to the names of applications for which you want to create the task.

   - Select the **Start application** or the **Stop application** option.

8. If you want to modify the default task settings, enable the **Open task details when creation is complete** option at the **Finish task creation** step. If you do not enable this option, the task is created with the default settings. You can modify the default settings later, at any time.

9. Click the **Finish** button.

   The task is created and displayed in the list of tasks.

10. Click the name of the created task to open the task properties window.

11. In the task properties window, specify the general task settings according to your needs, and then save the settings.

The task is created and configured.

To run the task, select it in the task list, and then click the **Start** button.

Replacing third-party security applications

Installation of Kaspersky security applications through Open Single Management Platform may require removal of third-party software that is incompatible with the application being installed. Open Single Management Platform provides several ways of removing the third-party applications.

Removing incompatible applications when configuring remote installation of an application

You can enable the **Uninstall incompatible applications automatically** option when you configure remote installation of a security application in the Protection deployment wizard. When this option is enabled, Open Single Management Platform removes incompatible applications before installing a security application on a managed device.

Removing incompatible applications through a dedicated task

To remove incompatible applications, use the *Uninstall application remotely* task. This task should be run on devices before the security application installation task. For example, in the installation task you can select **On completing another task** as the schedule type where the other task is *Uninstall application remotely*.

This method of uninstallation is useful when the security application installer cannot properly remove an incompatible application.

Removing applications or software updates remotely

You can remove applications or software updates on managed devices that run Linux remotely only by using Network Agent.

To remove applications or software updates remotely from selected devices:

1. In the main menu, go to **Asset management** → **Tasks**.

2. Click **Add**.

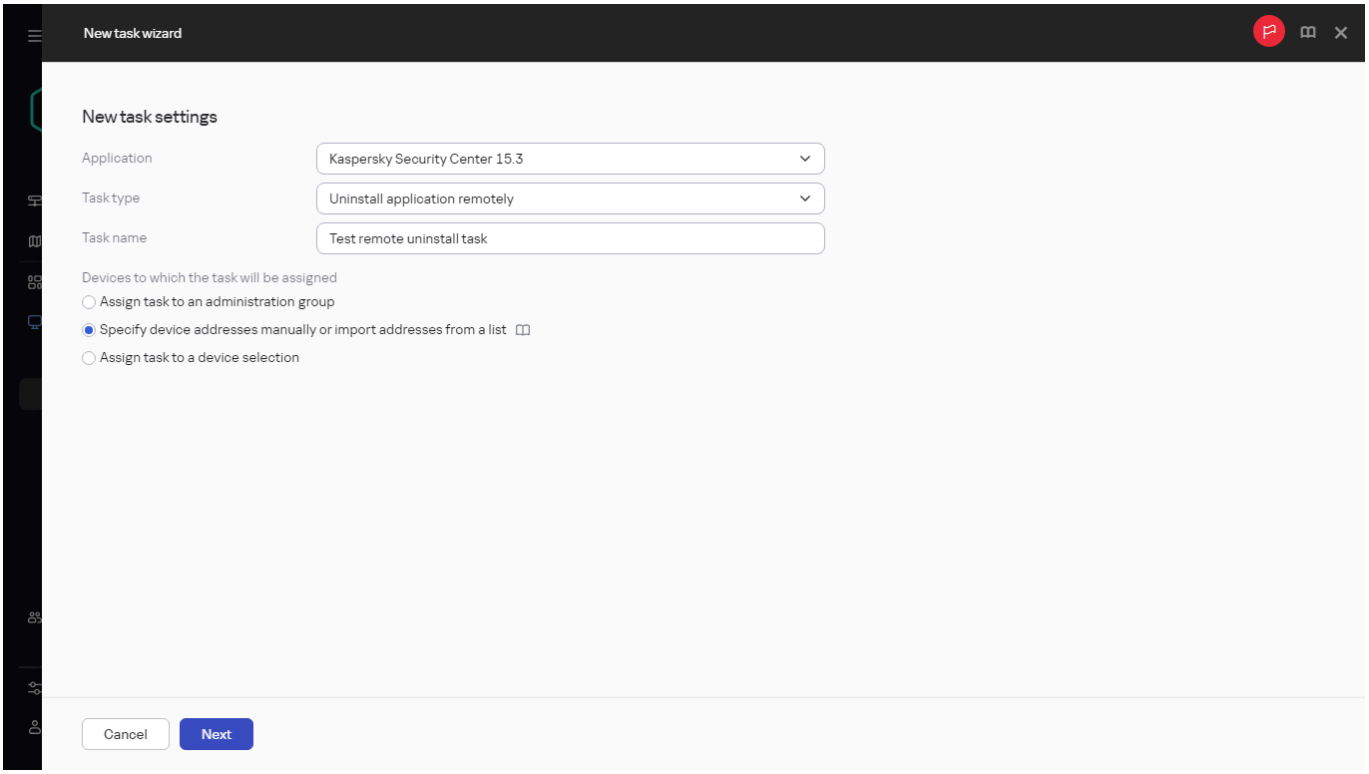
   The New task wizard starts. Proceed through the wizard by using the **Next** button.

3. In the **Application** drop-down list, select Open Single Management Platform.

4. In the **Task type** list, select the **Uninstall application remotely** task type.

5. In the **Task name** field, specify the name of the new task.

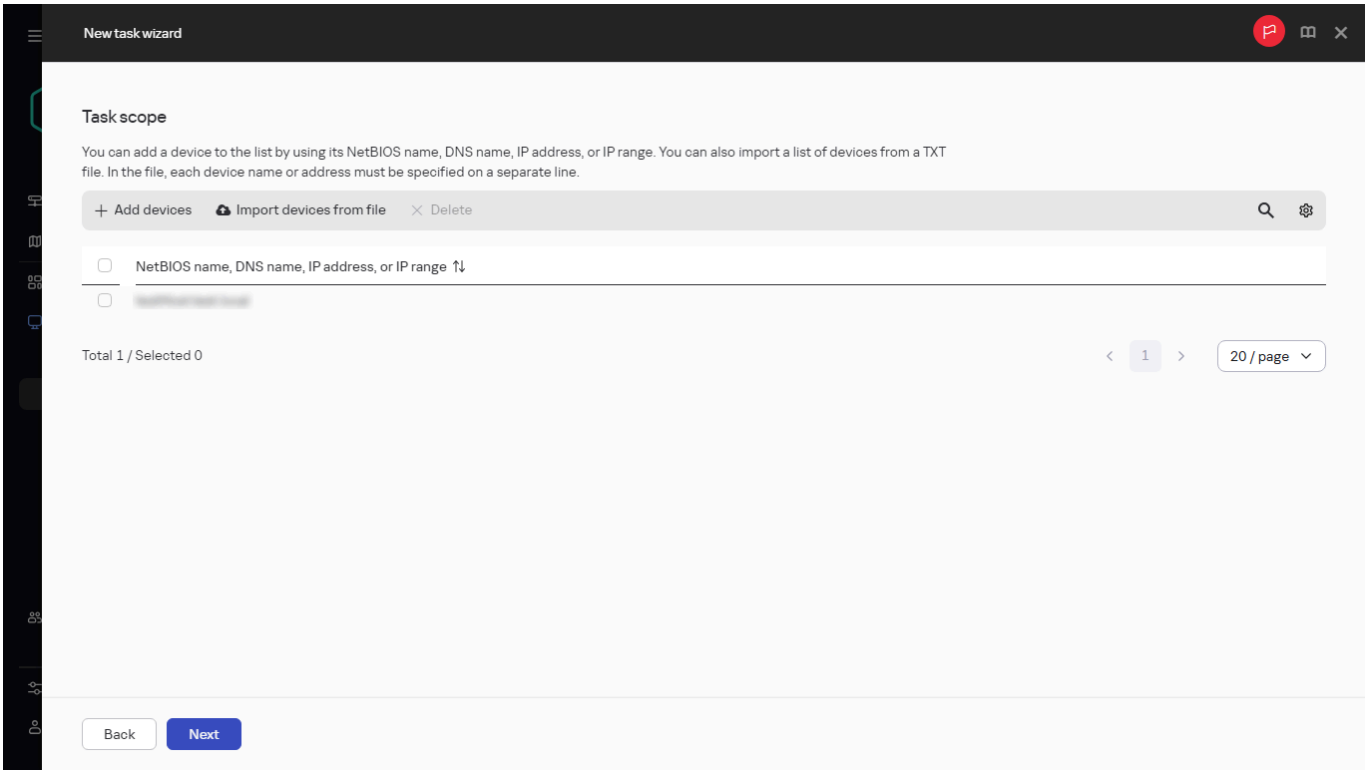
   A task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).



Creating the Uninstall application remotely task

6. Select the devices to which the task will be assigned.

   Go to the next step of the wizard.



Specifying the task scope

7. Select what kind of software you want to remove, and then select specific applications, updates, or patches that you want to remove:

- **Uninstall managed application**

  A list of Kaspersky applications is displayed. Select the application that you want to remove.

  Ensure that the **Use uninstallation password** policy setting is disabled for the managed application.

- **Uninstall incompatible application**

  A list of applications incompatible with Kaspersky security applications or Open Single Management Platform is displayed. Select the check boxes next to the applications that you want to remove.

- **Uninstall application from applications registry**

  By default, Network Agents send the Administration Server information about the applications installed on the managed devices. The list of installed applications is stored in the applications registry.

  To select an application from the applications registry:

  a. Click the **Application to uninstall** field, and then select the application that you want to remove.

  If you select Kaspersky Security Center Network Agent, when you run the task, the status *Completed successfully* shows that the process of removing started. If Kaspersky Security Center Network Agent is removed, the status does not change. If the task fails, the status changes to *Failed*.

  b. Specify the uninstallation options:

  - **Uninstallation mode**

    Select how you want to remove the application:

    - **Define uninstallation command automatically**

      If the application has an uninstallation command defined by the application vendor, Open Single Management Platform uses this command. We recommend that you select this option.

    - **Specify uninstallation command**

      Select this option if you want to specify your own command for the application uninstallation.

      We recommend that you first try to remove the application by using the **Define uninstallation command automatically** option. If the uninstallation through the automatically defined command fails, then use your own command.

      Type an installation command into the field, and then specify the following option:

      Use this command for uninstallation only if the default command was not autodetected

Open Single Management Platform checks whether or not the selected application has an uninstallation command defined by the application vendor. If the command is found, Open Single Management Platform will use it instead of the command specified in the **Command for application uninstallation** field.

We recommend that you enable this option.

- **Perform restart after successful application uninstallation**

  If the application requires the operating system to be restarted on the managed device after successful uninstallation, the operating system is restarted automatically.

- **Uninstall the specified application update, patch, or third-party application**

  A list of updates, patches, and third-party applications is displayed. Select the item that you want to remove.

  The displayed list is a general list of applications and updates, and it does not correspond to the applications and updates installed on the managed devices. Before selecting an item, we recommend that you ensure that the application or update is installed on the devices defined in the task scope. You can view the list of devices on which the application or update is installed, via the properties window.

  To view the list of devices:

  a. Click the name of the application or update.

     The properties window opens.

  b. Open the **Devices** section.

     You can also view the list of installed applications and updates in the device properties window.

8. Specify how client devices will download the Uninstallation utility:

- **Using Network Agent**

  The files are delivered to client devices by Network Agent installed on those client devices.

  If this option is disabled, the files are delivered using the Linux operating system tools.

  We recommend that you enable this option if the task has been assigned to devices that have Network Agents installed.

- **Using operating system resources through Administration Server**

  The option is obsolete. Use the **Using Network Agent** or **Using operating system resources through distribution points** option instead.

  The files are transmitted to client devices by using the Administration Server operating system tools. You can enable this option if no Network Agent is installed on the client device, but the client device is on the same network as the Administration Server.

- **Using operating system resources through distribution points**

  The files are transmitted to client devices by using operating system tools through distribution points. You can enable this option if there is at least one distribution point on the network.

  If the **Using Network Agent** option is enabled, the files are delivered by using operating system tools only if Network Agent tools are unavailable.

- **Maximum number of concurrent downloads**

  The maximum allowed number of client devices to which Administration Server can simultaneously transmit the files. The larger this number, the faster the application will be uninstalled, but the load on Administration Server is higher.

- **Maximum number of uninstallation attempts**

  If, when running the *Uninstall application remotely* task, Open Single Management Platform fails to uninstall an application on a managed device within the number of installer runs specified by the parameter, Open Single Management Platform stops delivering the Uninstallation utility to this managed device and does not start the installer on the device anymore.

  The **Maximum number of uninstallation attempts** parameter allows you to save the resources of the managed device, as well as reduce traffic (uninstallation, MSI file run, and error messages).

  Recurring task start attempts may indicate a problem on the device and which prevents uninstallation. The administrator should resolve the problem within the specified number of uninstallation attempts and then restart the task (manually or by a schedule).

  If uninstallation is not achieved eventually, the problem is considered unresolvable and any further task starts are seen as costly in terms of unnecessary consumption of resources and traffic.

  When the task is created, the attempts counter is set to `0`. Each run of the installer that returns an error on the device increments the counter reading.

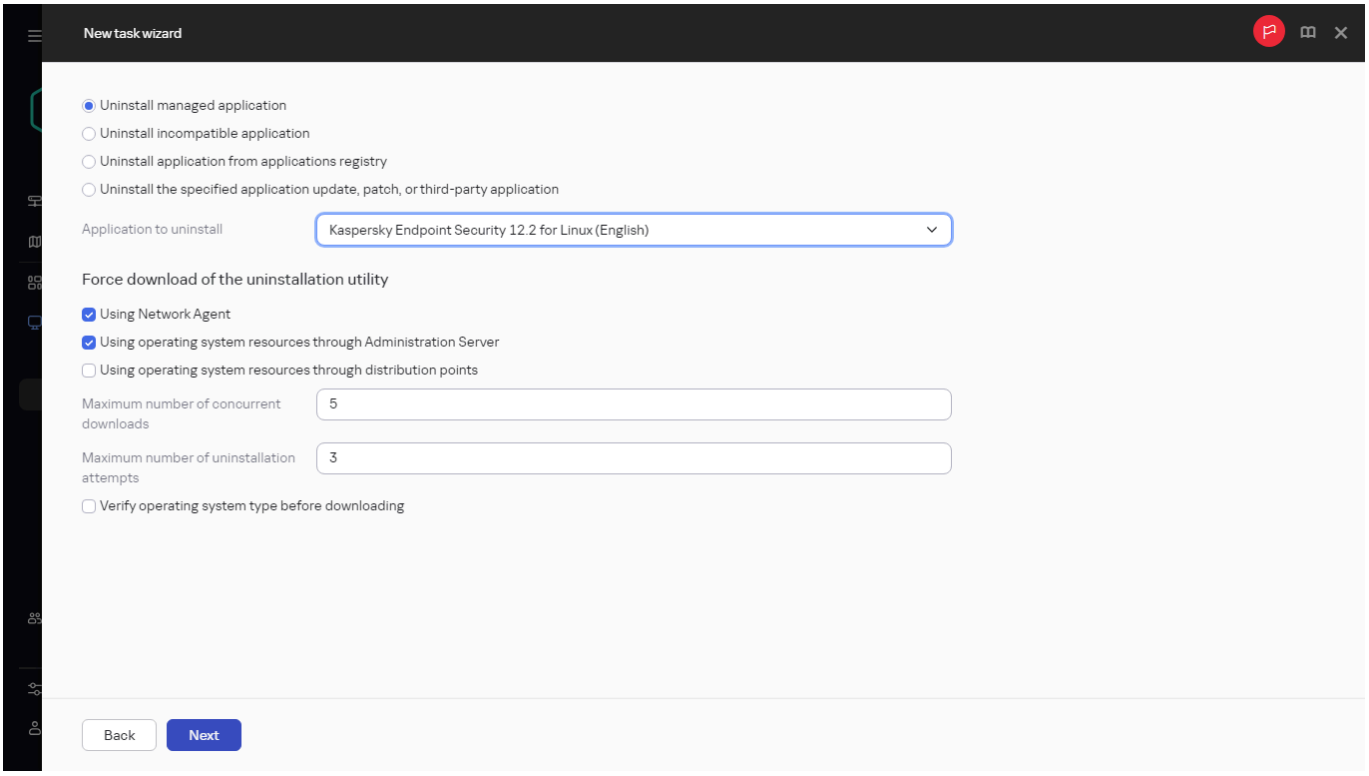
  If the number of attempts specified in the parameter has been exceeded and the device is ready for application uninstallation, you can increase the value of the **Maximum number of uninstallation attempts** parameter and start the task to uninstall the application. Alternatively, you can create a new *Uninstall application remotely* task.

- **Verify operating system type before downloading**

  Before transmitting the files to client devices, Open Single Management Platform checks if the Installation utility settings are applicable to the operating system of the client device. If the settings are not applicable, Open Single Management Platform does not transmit the files and does not attempt to install the application. For example, to install some application to devices of an administration group that includes devices running various operating systems, you can assign the installation task to the administration group, and then enable this option to skip devices that run an operating system other than the required one.

  When you remotely install Network Agent on unassigned devices, Open Single Management Platform or a distribution point automatically determines the device's operating system type, regardless of the **Verify operating system type before downloading** setting.

Go to the next step of the wizard.



Specifying the task settings

9. Specify the operating system restart settings:

- **Do not restart the device**

  Client devices are not restarted automatically after the operation. To complete the operation, you must restart a device (for example, manually or through a device management task). Information about the required restart is saved in the task results and in the device status. This option is suitable for tasks on servers and other devices where continuous operation is critical.

- **Restart the device**

  Client devices are always restarted automatically if a restart is required for completion of the operation. This option is useful for tasks on devices that provide for regular pauses in their operation (shutdown or restart).

- **Prompt user for action**

  The restart reminder is displayed on the screen of the client device, prompting the user to restart it manually. Some advanced settings can be defined for this option: text of the message for the user, the message display frequency, and the time interval after which a restart will be forced (without the user's confirmation). This option is most suitable for workstations where users must be able to select the most convenient time for a restart.

  By default, this option is selected.

  - **Repeat prompt every (min)**
  - **Restart after (min)**

- **Force closure of applications in blocked sessions**

  Running applications may prevent a restart of the client device. For example, if a document is being edited in a word processing application and is not saved, the application does not allow the device to restart.

  If this option is enabled, such applications on a locked device are forced to close before the device restart. As a result, users may lose their unsaved changes.

  If this option is disabled, a locked device is not restarted. The task status on this device states that a device restart is required. Users have to manually close all applications running on locked devices and restart these devices.

  By default, this option is disabled.

Go to the next step of the wizard.

10. If necessary, add the accounts that will be used to start the remote uninstallation task:

- **No account required (Network Agent installed)**

  If this option is selected, you do not have to specify the account under which the application installer will be run. The task will run under the account under which the Administration Server service is running.

  If Network Agent has not been installed on client devices, this option is not available.

- **Account required (Network Agent is not used)**

  Select this option if Network Agent is not installed on the devices for which you assign the *Uninstall application remotely* task. In this case, you can specify a user account or an SSH certificate to uninstall the application.

  - **Local Account**. If this option is selected, specify the user account under which the application installer will be run. Click the **Add** button, select **Local Account**, and then specify the user account credentials.

    You can specify multiple user accounts if, for example, none of them have all the required rights on all devices for which you assign the task. In this case, all added accounts are used for running the task, in consecutive order, top-down.

  - **SSH certificate**. If you want to uninstall an application from a Linux-based client device, you can specify an SSH certificate instead of a user account. Click the **Add** button, select **SSH certificate**, and then specify the private and public keys of the certificate.

    To generate a private key, you can use the ssh-keygen utility. Note that Open Single Management Platform supports the PEM format of private keys, but the ssh-keygen utility generates SSH keys in the OPENSSH format by default. The OPENSSH format is not supported by Open Single Management Platform. To create a private key in the supported PEM format, add the -m PEM option in the ssh-keygen command.

    For example:

    ```
    ssh-keygen -m PEM -t rsa -b 4096 -C "<user email>"
    ```

11. At the **Finish task creation** step of the wizard, enable the **Open task details when creation is complete** option to modify the default task settings.

    If you do not enable this option, the task will be created with the default settings. You can modify the default settings later.

12. Click the **Finish** button.

    The wizard creates the task. If you enabled the **Open task details when creation is complete** option, the task properties window automatically opens. In this window, you can specify the general task settings and, if required, change the settings specified during task creation.

    > You can also open the task properties window by clicking the name of the created task in the list of tasks.

    The task is created, configured, and displayed in the list of tasks at **Asset management** → **Tasks**.

13. To run the task, select it in the task list, and then click the **Start** button.

    You can also set a task start schedule on the **Schedule** tab of the task properties window.

    For a detailed description of scheduled start settings, refer to the general task settings.

After the task is completed, the selected application is removed from the selected devices.

Remote uninstallation issues

Sometimes remote uninstallation of third-party applications may finish with the following warning: "Remote uninstallation has finished on this device with warnings: Application for removal is not installed." This issue occurs when the application to be uninstalled has already been uninstalled or was installed only for an individual user. Applications installed for an individual user (also referred to as per-user applications) become invisible and cannot be uninstalled remotely if the user is not logged in.

This behavior differs from applications intended for use by multiple users on the same device (also referred to as per-device applications). Per-device applications are visible and accessible to all users of the device.

Therefore, per-user applications must be uninstalled only when the user is logged in.

Source of information about installed applications

The Network Agent retrieves information about software installed on Windows devices from the following registry keys:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall

  Contains information about applications installed for all users.

- HKEY_LOCAL_MACHINE\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall

  Contains information about applications installed for all users.

- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Uninstall

  Contains information about applications installed for the current user.

- HKEY_USER<...>\Software\Microsoft\Windows\CurrentVersion\Uninstall

  Contains information about applications installed for specific users.

Preparing a device running SUSE Linux Enterprise Server 15 for installation of Network Agent

Prior to the installation of Network Agent on a device running SUSE Linux Enterprise Server 15, you must perform two preparation procedures—the one in the instructions below and general preparation steps for any Linux device.

To install Network Agent on a device with the SUSE Linux Enterprise Server 15 operating system:

Before the Network Agent installation, run the following command:

```
sudo zypper install insserv-compat
```

This enables you to install the insserv-compat package and configure Network Agent properly.

Run the `rpm -q insserv-compat` command to check whether the package is already installed.

If your network includes a lot of devices running SUSE Linux Enterprise Server 15, you can use the special software for configuring and managing the company infrastructure. By using this software, you can automatically install the insserv-compat package on all necessary devices at once. For example, you can use Puppet, Ansible, Chef, you can make your own script—use any method that is convenient for you.

If the device does not have the GPG signing keys for SUSE Linux Enterprise, you may encounter the following warning: `Package header is not signed!` Select the `i` option to ignore the warning.

After preparing the Linux device, you can install Network Agent remotely, in interactive mode, or in silent mode.

Preparing a Windows device for remote installation

Remote installation of the application on the client device may return an error for the following reasons:

- The task has already been successfully performed on this device.

  In this case, the task does not have to be performed again.

- When a task was started, the device was shut down.

  In this case, turn on the device, and then restart the task.

- There is no connection between the Administration Server and the Network Agent installed on the client device.

  To determine the cause of the problem, use the utility designed for remote diagnostics of client devices (klactgui).

- If Network Agent is not installed on the device, the following issues may occur during remote installation:
  - Network errors
  - Misconfigured operating system
  - Incorrectly configured account rights in the remote installation task

  To avoid issues that may occur during installation of the application on a client device without Network Agent installed, you must force the installation of selected installation packages by using the remote installation task of Open Single Management Platform—provided that each device has a user account with local administrator rights.

Previously, the riprep utility was used to prepare a Windows device for remote installation. This is now considered an outdated method for configuring operating systems. The riprep utility is not recommended for use on operating systems newer than Windows XP and Windows Server 2003 R2.

Forced installation can also be applied if devices cannot be directly accessed by Administration Server. For example, if the devices are on isolated networks or on a local network, while Administration Server is in the DMZ. In such cases, a distribution point is required for deployment to such devices.

Using distribution points as local installation centers may also be useful when performing installation on devices in subnets communicating with Administration Server via a low-capacity channel while a broader channel is available between devices in the same subnet.

In case of initial deployment, Network Agent is not installed. Therefore, in the settings of the remote installation task, you cannot select distribution of files required for application installation by using Network Agent. You can only choose to distribute files by using operating system resources through Administration Server or distribution points.

You should specify an account that has access to the admin$ share in the settings of the remote installation task.

You can specify target devices either explicitly (with a list), by selecting the Open Single Management Platform administration group to which they belong, or by creating a selection of devices based upon a specific criterion. The installation start time is defined by the task schedule. If the **Run missed tasks** setting is enabled in the task properties, the task can be run either immediately after target devices are turned on or when they are moved to the target administration group.

Forced installation consists of delivering installation packages to target devices, subsequent copying of files to the admin$ resource on each of the target devices, and remote registration of supporting services on those devices. Delivery of installation packages to target devices is performed through the Open Single Management Platform feature that ensures network interaction. The following conditions must be met in this case:

- Target devices are accessible from the distribution point with the Windows operating system, from which remote installation to client devices is to be carried out and this distribution point is selected for the target devices.
- Name resolution for target devices functions properly on the network.
- The administrative shares (admin$) remain enabled on target devices.
- The following system services are running on target devices:
  - Server (LanmanServer)

    By default, this service is running.

- DCOM Server Process Launcher (DcomLaunch)
- RPC Endpoint Mapper (RpcEptMapper)
- Remote Procedure Call (RpcSs)

- Port TCP 445 is open on target devices, to enable remote access through Windows tools.

  TCP 139, UDP 137, and UDP 138 are used by older protocols and are no longer necessary for current applications.

  Dynamic outbound access ports must be allowed on the firewall, for connections from distribution points to target devices.

- The Active Directory domain policy security settings are allowed to provide the operation of the NTLM protocol during the deployment of Network Agent.

- On target devices running Microsoft Windows XP, Simple File Sharing mode is disabled.

- On target devices, the access sharing and security model are set as *Classic – local users authenticate as themselves*. It can in no way be *Guest only – local users authenticate as Guest*.

- Target devices are members of the domain, or uniform accounts with administrator rights are created on target devices in advance.

To deploy Network Agent or other applications successfully to a device that is not joined to a Windows Server 2003 or later Active Directory domain, you must disable remote UAC on that device. Remote UAC is one of the reasons that prevent local administrative accounts from accessing admin$, which is necessary for forced deployment of Network Agent or other applications. Disabling remote UAC does not affect local UAC.

During installation on new devices that have not yet been allocated to any of the Open Single Management Platform administration groups, you can open the remote installation task properties and specify the administration group to which devices will be moved after Network Agent installation.

When creating a group task, keep in mind that each group task affects all devices in all nested groups within a selected group. Therefore, you must avoid duplicating installation tasks in subgroups.

A simplified way to create tasks for forced installation of applications is automatic installation. To do this, you must open the administration group properties, open the list of installation packages, and then select the ones that must be installed on devices in this group. As a result, the selected installation packages will be automatically installed on all devices in this group and all of its subgroups. The time interval over which the packages will be installed depends on the network throughput and the total number of networked devices.

You can use several distribution points to reduce the load during the delivery of installation packages to target devices. Note that this installation method places a significant load on devices acting as distribution points. If you use distribution points, you have to make sure that they are present in each of the isolated subnets hosting target devices.

The free disk space in the partition with the %ALLUSERSPROFILE%\Application Data\KasperskyLab\adminkit folder must exceed, by many times, the total size of the distribution packages of installed applications.

Configuring Kaspersky applications

This section contains information about manual configuration of policies and tasks, about user roles, about building an administration group structure and hierarchy of tasks.

Scenario: Configuring network protection

Create and configure policies and tasks required for your network.

Prerequisites

Before you start, make sure that you have done the following:

- Installed Kaspersky Security Center Administration Server
- Installed OSMP Console
- Completed the Open Single Management Platform main installation scenario

Configuring network protection proceeds in stages:

1. **Setup and propagation of Kaspersky application policies and policy profiles**

   To configure and propagate settings for Kaspersky applications installed on the managed devices, you can use two different security management approaches—device-centric or user-centric. These two approaches can also be combined.

2. **Configuring tasks for remote management of Kaspersky applications**

   Manually create and configure the following policies and tasks in the **Managed devices** administration group:

   - Policy of Kaspersky Endpoint Security
   - Group task for updating Kaspersky Endpoint Security
   - Policy of Network Agent

   How-to instructions: Setting up the group task for updating Kaspersky Endpoint Security.

   If necessary, create additional tasks to manage the Kaspersky applications installed on the client devices.

3. **Evaluating and limiting the event load on the database**

   Information about events that occur during the operation of managed applications is transferred from a client device and registered in the Administration Server database. To reduce the load on the Administration Server, evaluate and limit the maximum number of events that can be stored in the database.

   How-to instructions: Setting the maximum number of events.

Results

Upon completion of this scenario, your network will be protected by configuration of Kaspersky applications, tasks, and events received by the Administration Server:

- The Kaspersky applications are configured according to the policies and policy profiles.
- The applications are managed through a set of tasks.
- The maximum number of events that can be stored in the database is set.

When the network protection configuration is complete, you can proceed to configuring regular updates to Kaspersky databases and applications.

About device-centric and user-centric security management approaches

You can manage security settings from the standpoint of device features and from the standpoint of user roles. The first approach is called *device-centric security management* and the second is called *user-centric security management*. To apply different application settings to different devices, you can use either or both types of management in combination.

Device-centric security management enables you to apply different security application settings to managed devices depending on device-specific features. For example, you can apply different settings to devices allocated in different administration groups.

User-centric security management enables you to apply different security application settings to different user roles. You can create several user roles, assign an appropriate user role to each user, and define different application settings to the devices owned by users with different roles. For example, you may want to apply different application settings to devices of accountants and human resources (HR) specialists. As a result, when user-centric security management is implemented, each department—accounts department and HR department—has its own settings configuration for Kaspersky applications. A settings configuration defines which application settings can be changed by users and which are forcibly set and locked by the administrator.

By using user-centric security management, you can apply specific application settings to individual users. This may be required when an employee has a unique role in the company or when you want to monitor security issues related to devices of a specific person. Depending on the role of this employee in the company, you can expand or limit the rights of this person to change application settings. For example, you might want to expand the rights of a system administrator who manages client devices in a local office.

You can also combine the device-centric and user-centric security management approaches. For example, you can configure a specific application policy for each administration group, and then create policy profiles for one or several user roles of your enterprise. In this case, the policies and policy profiles are applied in the following order:

1. The policies created for device-centric security management are applied.
2. They are modified by the policy profiles according to the policy profile priorities.
3. The policies are modified by the policy profiles associated with user roles.

Policy setup and propagation: Device-centric approach

When you complete this scenario, the applications will be configured on all of the managed devices in accordance with the application policies and policy profiles that you define.

Prerequisites

Before you start, make sure that you have installed Kaspersky Security Center Administration Server and OSMP Console. You might also want to consider user-centric security management as an alternative or additional option to the device-centric approach. Learn more about two management approaches.

Stages

The scenario of device-centric management of Kaspersky applications consists of the following steps:

1. **Configuring application policies**

   Configure settings for Kaspersky applications installed on the managed devices by creating a policy for each application. The set of policies will be propagated to the client devices.

   If you have a hierarchical structure of several Administration Servers and/or administration groups, the secondary Administration Servers and child administration groups inherit the policies from the primary Administration Server by default. You can force the inheritance by the child groups and secondary Administration Servers to prohibit any modifications of the settings configured in the upstream policy. If you want only part of the settings to be forcibly inherited, you can lock them in the upstream policy. The rest unlocked settings will be available for modification in the downstream policies. The created hierarchy of policies will allow you to effectively manage devices in the administration groups.

   How-to instructions: Creating a policy

2. **Creating policy profiles (optional)**

   If you want devices within a single administration group to run under different policy settings, create policy profiles for those devices. A policy profile is a named subset of policy settings. This subset is distributed on target devices together with the policy, supplementing it under a specific condition called the *profile activation condition*. Profiles only contain settings that differ from the "basic" policy, which is active on the managed device.

   By using profile activation conditions, you can apply different policy profiles, for example, to the devices having a specific hardware configuration or marked with specific tags. Use tags to filter devices that meet specific criteria. For example, you can create a tag called *CentOS*, mark all devices running CentOS operating system with this tag, and then specify this tag as an activation condition for a policy profile. As a result, Kaspersky applications installed on all devices running CentOS will be managed by their own policy profile.

   How-to instructions:

   - Creating a policy profile
   - Creating a policy profile activation rule

3. **Propagating policies and policy profiles to the managed devices**

   By default, the Administration Server automatically synchronizes with managed devices every 15 minutes. During the synchronization, the new or changed policies and policy profiles are propagated to the managed devices. You can circumvent auto-synchronization and run the synchronization manually by using the Force synchronization command. When synchronization is complete, the policies and policy profiles are delivered and applied to the installed Kaspersky applications.

You can check whether the policies and policy profiles were delivered to a device. Open Single Management Platform specifies the delivery date and time in the properties of the device.

How-to instructions: Forced synchronization

Results

When the device-centric scenario is complete, the Kaspersky applications are configured according to the settings specified and propagated through the hierarchy of policies.

The configured application policies and policy profiles will be applied automatically to the new devices added to the administration groups.

Policy setup and propagation: User-centric approach

This section describes the scenario of user-centric approach to the centralized configuration of Kaspersky applications installed on the managed devices. When you complete this scenario, the applications will be configured on all of the managed devices in accordance with the application policies and policy profiles that you define.

Prerequisites

Before you start, make sure that you have successfully installed Kaspersky Security Center Administration Server and OSMP Console, and completed the main deployment scenario. You might also want to consider device-centric security management as an alternative or additional option to the user-centric approach. Learn more about two management approaches.

Process

The scenario of user-centric management of Kaspersky applications consists of the following steps:

1. **Configuring application policies**

   Configure settings for Kaspersky applications installed on the managed devices by creating a policy for each application. The set of policies will be propagated to the client devices.

   If you have a hierarchical structure of several Administration Servers and/or administration groups, the secondary Administration Servers and child administration groups inherit the policies from the primary Administration Server by default. You can force the inheritance by the child groups and secondary Administration Servers to prohibit any modifications of the settings configured in the upstream policy. If you want only part of the settings to be forcibly inherited, you can lock them in the upstream policy. The rest unlocked settings will be available for modification in the downstream policies. The created hierarchy of policies will allow you to effectively manage devices in the administration groups.

   How-to instructions: Creating a policy

2. **Specifying owners of the devices**

   Assign the managed devices to the corresponding users.

   How-to instructions: Assigning a user as a device owner

3. **Defining user roles typical for your enterprise**

   Think about different kinds of work that the employees of your enterprise typically perform. You must divide all employees in accordance with their roles. For example, you can divide them by departments, professions, or positions. After that you will need to create a user role for each group. Keep in mind that each user role will have its own policy profile containing application settings specific for this role.

4. **Creating user roles**

   Create and configure a user role for each group of employees that you defined on the previous step or use the predefined user roles. The user roles will contain set of rights of access to the application features.

   How-to instructions: Creating a user role

5. **Defining the scope of each user role**

   For each of the created user roles, define users and/or security groups and administration groups. Settings associated with a user role apply only to devices that belong to users who have this role, and only if these devices belong to groups associated with this role, including child groups.

   How-to instructions: Editing the scope of a user role

6. **Creating policy profiles**

   Create a policy profile for each user role in your enterprise. The policy profiles define which settings will be applied to the applications installed on users' devices depending on the role of each user.

   How-to instructions: Creating a policy profile

7. **Associating policy profiles with the user roles**

   Associate the created policy profiles with the user roles. After that: the policy profile becomes active for a user that has the specified role. The settings configured in the policy profile will be applied to the Kaspersky applications installed on the user's devices.

   How-to instructions: Associating policy profiles with roles

8. **Propagating policies and policy profiles to the managed devices**

   By default, Open Single Management Platform automatically synchronizes the Administration Server with the managed devices every 15 minutes. During the synchronization, the new or changed policies and policy profiles are propagated to the managed devices. You can circumvent auto-synchronization and run the synchronization manually by using the Force synchronization command. When synchronization is complete, the policies and policy profiles are delivered and applied to the installed Kaspersky applications.

   You can check whether the policies and policy profiles were delivered to a device. Open Single Management Platform specifies the delivery date and time in the properties of the device.

   How-to instructions: Forced synchronization

Results

When the user-centric scenario is complete, the Kaspersky applications are configured according to the settings specified and propagated through the hierarchy of policies and policy profiles.

For a new user, you will have to create a new account, assign the user one of the created user roles, and assign the devices to the user. The configured application policies and policy profiles will be automatically applied to the devices of this user.

About switching Network Agent to other Administration Servers

Open Single Management Platform provides the option of switching Network Agent on a client device to other Administration Servers if the following settings of the network have been changed:

- **Condition for DHCP server address**—The IP address of the network Dynamic Host Configuration Protocol (DHCP) server has changed.

  The **Condition for DHCP server address** setting is currently not available.

- **Condition for default connection gateway address**—The address of the main network gateway has changed.
- **Condition for DNS domain**—The DNS suffix of the subnet has changed.
- **Condition for DNS server address**—The IP address of the network DNS server has changed.
- **Condition for WINS server address**—The IP address of the network WINS server has changed. This setting is available only for devices running Windows.
- **Condition for name resolvability**—The DNS or NetBIOS name of the client device has changed.
- **Condition for subnet**—Changes the subnet address and mask.
- **Condition for Windows domain accessibility**—Changes the status of the Windows domain to which the client device is connected. This setting is available only for devices running Windows.
- **Condition for SSL connection address accessibility**—The client device can or cannot (depending on the option that you select) establish an SSL connection with a specified server (name:port). You must use a dedicated server with software capable of handling high load and frequent Network Agent requests for the connection. Do not specify the Administration Server's port. For each server, you can additionally specify an SSL certificate. In this case, the Network Agent verifies the server certificate in addition to checking the capability of an SSL connection. If the certificate does not match, the connection fails.

The initial settings of the Network Agent connection to Administration Server are defined when installing the Network Agent. Afterwards, if rules for switching the Network Agent to other Administration Servers have been created, the Network Agent responds to changes in the network settings as follows:

- If the network settings comply with one of the rules created, Network Agent connects to the Administration Server specified in this rule. Applications installed on client devices switch to out-of-office policies, provided such behavior is enabled by a rule.
- If none of the rules apply, Network Agent reverts to the default settings of connection to the Administration Server specified during the installation. Applications installed on client devices switch back to active policies.
- If the Administration Server is not accessible, Network Agent uses out-of-office policies.

Network Agent switches to the out-of-office policy only if the **Enable out-of-office mode when Administration Server is not available** option is enabled in the Network Agent policy settings.

The settings of Network Agent connection to Administration Server are saved in a connection profile. In the connection profile, you can create rules for switching client devices to out-of-office policies, and you can configure the profile so that it could only be used for downloading updates.

Creating a connection profile for out-of-office users

To create a profile for connecting Network Agent to Administration Server for out-of-office users:

1. If you want to create a connection profile for a group of managed devices, open the Network Agent policy of this group. To do this, do the following:

   a. In the main menu, go to **Asset management** → **Policies**.

   b. Select the administration group for which the connection profile is to be created:

      - For the root group, you can proceed to the next step.

      - For a subgroup, click the **Show hierarchy** button at the top of the page, and then in the window that opens, click the name of the required subgroup.

      The path to the selected group is displayed at the top of the page. If necessary, you can click a link with the administration group name to go to the group. By default, the last link in the path is inactive. Also, you can close the hierarchy window by clicking the **Hide hierarchy** button.

   c. Add the Network Agent policy for the group of managed devices. If you have already created it, click the Network Agent policy name to open the policy properties.

2. If you want to create a connection profile for a specific managed device, do the following:

   a. In the main menu, go to **Asset management** → **Assets (Devices)**.

   b. Click the name of the managed device.

   c. In the managed device properties window that opens, go to the **Applications** tab.

   d. Click the name of the Network Agent policy to which only the selected managed device applies.

3. In the properties window that opens, go to **Application settings** → **Connectivity** → **Connection profiles.**

4. In the **Administration Server connection profiles** section, click the **Add** button.

   By default, the list of connection profiles contains the <Offline mode> and <Home Administration Server> profiles. Profiles cannot be edited or removed.

   The <Offline mode> profile does not specify any Server for connection. Therefore, Network Agent, when switched to that profile, does not attempt to connect to any Administration Server while applications installed on client devices run under out-of-office policies. The <Offline mode> profile can be used if devices are disconnected from the network.

   The <Home Administration Server> profile specifies the connection for the Administration Server that was selected during Network Agent installation. The <Home Administration Server> profile is applied when a device is reconnected to the home Administration Server after it was running on an external network for some time.

5. In the **Configure profile** window that opens, configure the connection profile:

- **Profile name**

  In the entry field you can view or change the connection profile name.

- **Administration Server address**

  Address of the Administration Server to which the client device must connect during profile activation.

- **Port number**

  Port number that is used for connection.

- **SSL port**

  Port number for connection if using the SSL protocol.

- **Use SSL connection**

  If this option is enabled, the connection is established through a secure port, by using SSL protocol.

  By default, this option is enabled. We recommend that you do not disable this option so your connection remains secured.

- Select the **Use proxy server** option if you want to use a proxy server when connecting to the internet. If this option is selected, fields are available for entering settings. Specify the following settings for a proxy server connection:

  - **Address**

    Address of the proxy server used for Open Single Management Platform connection to the internet.

  - **Port number**

    Number of the port through which Open Single Management Platform proxy connection will be established.

  - **Proxy server authentication**

    If this check box is selected, in the entry fields you can specify the credentials for proxy server authentication.

  - **User name**

    User account under which connection to the proxy server is established (this field is available if the **Proxy server authentication** check box is selected).

- **Password**

  Password set by the user under whose account the proxy server connection is established (this field is available if the **Proxy server authentication** check box is selected).

  To see the entered password, click and hold the **Show** button for as long as you require.

- **Connection gateway address**

  Address of the gateway through which client devices connect to the Administration Server.

- **Enable out-of-office mode when Administration Server is not available**

  Select this check box to allow the applications installed on a client device to use policy profiles for devices in out-of-office mode, as well as out-of-office policies, at any connection attempt if the Administration Server is not available. If no out-of-office policy has been defined for the application, the active policy will be used.

  If this option is disabled, applications will use active policies.

  By default, this check box is cleared.

- **Use to receive updates only**

  If this option is enabled, the profile will only be used for downloading updates by applications installed on the client device. For other operations, connection to the Administration Server will be established with the initial connection settings defined during Network Agent installation.

  By default, this option is enabled.

- **Synchronize connection settings with the Administration Server settings specified in this profile**

  If this option is enabled, Network Agent connects to Administration Server using the settings specified in the profile properties.

  If this option is disabled, Network Agent connects to Administration Server using the original settings that have been specified during installation.

  This option is available if the **Use to receive updates only** option is disabled.

  By default, this option is disabled.

A profile for connecting Network Agent to Administration Server is created for out-of-office users. When Network Agent connects to Administration Server by using this profile, applications installed on the client device will use policies for devices in out-of-office mode or out-of-office policies.

Policies and policy profiles

In OSMP Console, you can create policies for Kaspersky applications. This section describes policies and policy profiles, and provides instructions for creating and modifying them.

About policies and policy profiles

A *policy* is a set of Kaspersky application settings that are applied to an administration group and its subgroups. You can install several Kaspersky applications on the devices of an administration group. Open Single Management Platform provides a single policy for each Kaspersky application in an administration group. A policy has one of the following statuses:

The status of the policy

Status	Description
Active	The current policy that is applied to the device. Only one policy may be active for a Kaspersky application in each administration group. Devices apply the settings values of an active policy for a Kaspersky application.
Inactive	A policy that is not currently applied to a device.
Out-of-office	If this option is selected, the policy becomes active when the device leaves the corporate network.

Policies function according to the following rules:

- Multiple policies with different values can be configured for a single application.
- Only one policy can be active for the current application.
- A policy can have child policies.

Generally, you can use policies as preparations for emergency situations, such as a virus attack. For example, if there is an attack via flash drives, you can activate a policy that blocks access to flash drives. In this case, the current active policy automatically becomes inactive.

In order to prevent maintaining multiple policies, for example, when different occasions assume changing of several settings only, you may use policy profiles.

A *policy profile* is a named subset of policy settings values that replaces the settings values of a policy. A policy profile affects the effective settings formation on a managed device. *Effective settings* are a set of policy settings, policy profile settings, and local application settings that are currently applied for the device.

Policy profiles function according to the following rules:

- A policy profile takes effect when a specific activation condition occurs.
- Policy profiles contain values of settings that differ from the policy settings.
- Activation of a policy profile changes the effective settings of the managed device.
- A policy can include a maximum of 100 policy profiles.

About lock and locked settings

Each policy setting has a lock button icon (🔒). The table below shows lock button statuses:

Lock button statuses

Status	Description
Undefined	If an open lock is displayed next to a setting and the toggle button is disabled, the setting is not specified in the policy. A user can change these settings in the managed application interface. These type of settings are called *unlocked*.
Enforce	If a closed lock is displayed next to a setting and the toggle button is enabled, the setting is applied to the devices where the policy is enforced. A user cannot modify the values of these settings in the managed application interface. These type of settings are called *locked*.

> We highly recommend that you close locks for the policy settings that you want to apply on the managed devices. The unlocked policy settings can be reassigned by Kaspersky application settings on a managed device.

You can use a lock button for performing the following actions:

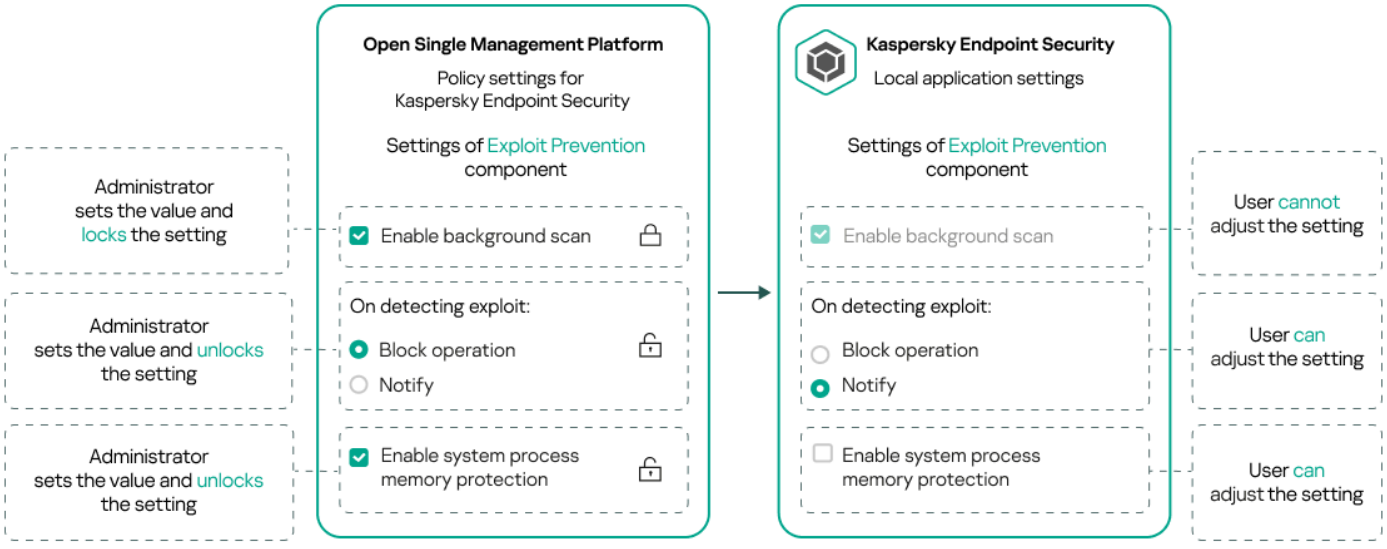
- Locking settings for an administration subgroup policy
- Locking settings of a Kaspersky application on a managed device

Thus, a locked setting is used for implementing effective settings on a managed device.

A process of effective settings implementation includes the following actions:

- Managed device applies settings values of Kaspersky application.
- Managed device applies locked settings values of a policy.

A policy and managed Kaspersky application contain the same set of settings. When you configure policy settings, the Kaspersky application settings change values on a managed device. You cannot adjust locked settings on a managed device (see the figure below):



Locks and Kaspersky application settings

Inheritance of policies and policy profiles

This section provides information about the hierarchy and inheritance of policies and policy profiles.

Hierarchy of policies

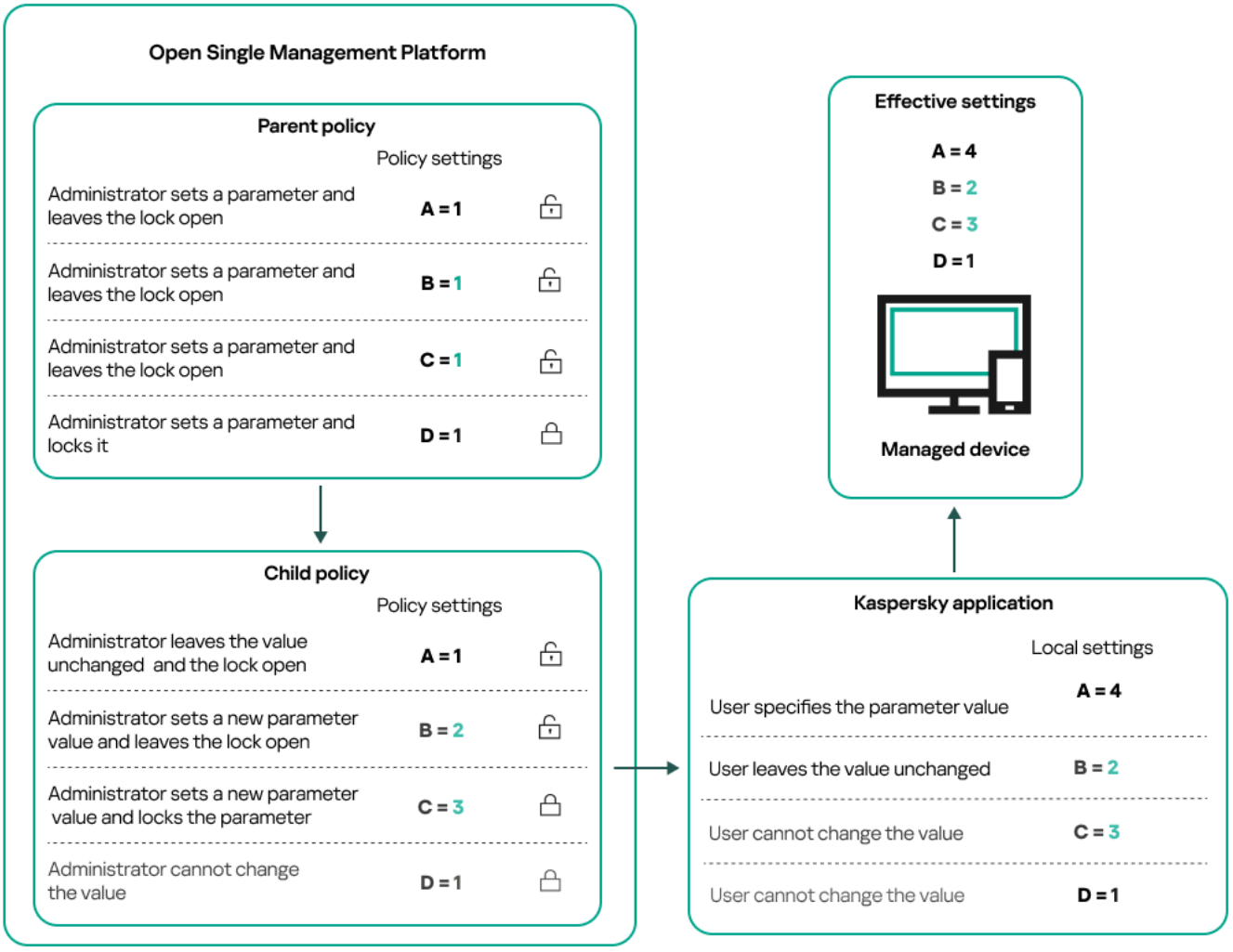
If different devices need different settings, you can organize devices into administration groups.

You can specify a policy for a single administration group. Policy settings can be *inherited*. Inheritance means receiving policy settings values in subgroups (child groups) from a policy of a higher-level (parent) administration group.

Hereinafter, a policy for a parent group is also referred to as a *parent policy*. A policy for a subgroup (child group) is also referred to as a *child policy*.

By default, at least one managed devices group exists on Administration Server. If you want to create custom groups, they are created as subgroups (child groups) within the managed devices group.

Policies of the same application act on each other, according to a hierarchy of administration groups. Locked settings from a policy of a higher-level (parent) administration group will reassign policy settings values of a subgroup (see the figure below).

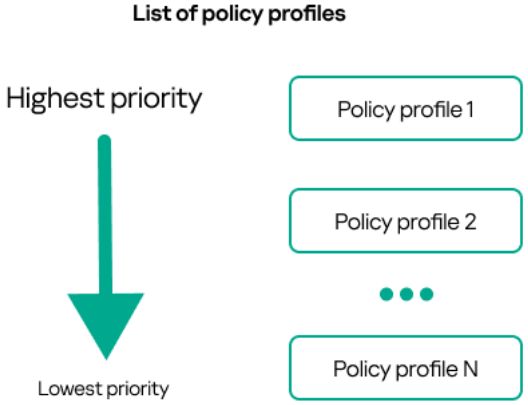


Hierarchy of policies

Policy profiles in a hierarchy of policies

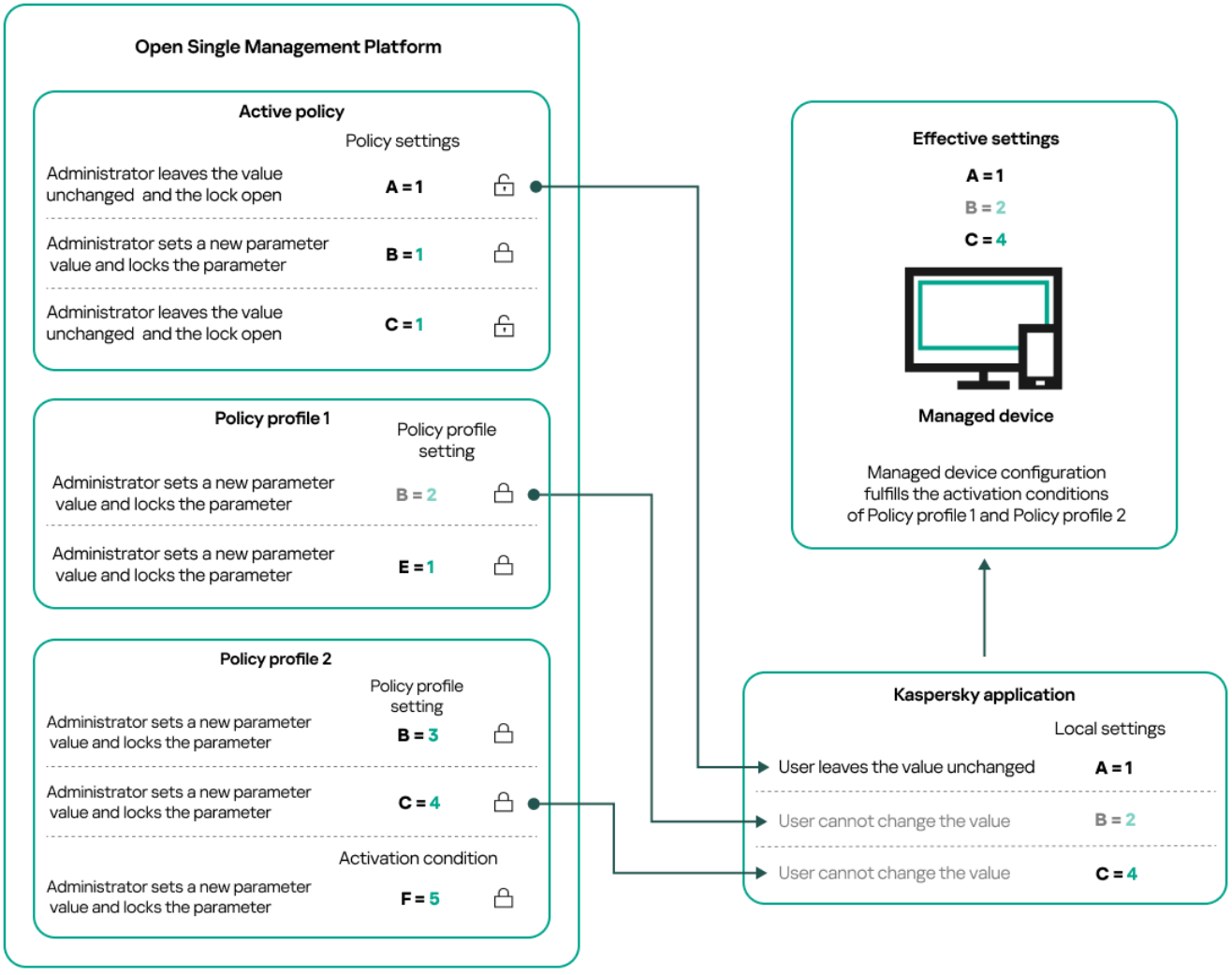
Policy profiles have the following priority assignment conditions:

- A profile's position in a policy profile list indicates its priority. You can change a policy profile priority. The highest position in a list indicates the highest priority (see the figure below).



Priority definition of a policy profile

- Activation conditions of policy profiles do not depend on each other. Several policy profiles can be activated simultaneously. If several policy profiles affect the same setting, the device takes the setting value from the policy profile with the highest priority (see the figure below).

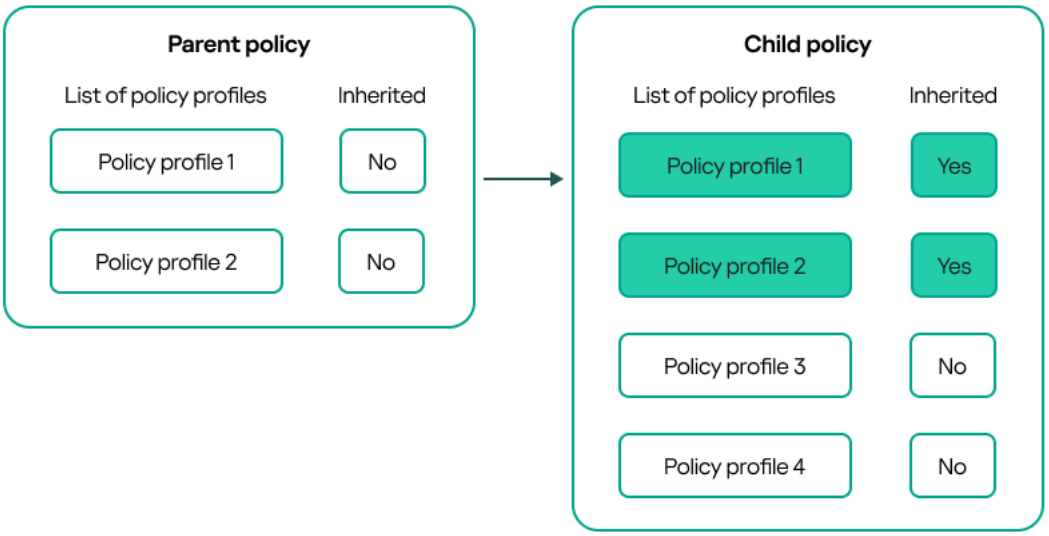


Managed device configuration fulfills activation conditions of several policy profiles

Policy profiles in a hierarchy of inheritance

Policy profiles from different hierarchy level policies comply with the following conditions:

- A lower-level policy inherits policy profiles from a higher-level policy. A policy profile inherited from a higher-level policy obtains higher priority than the original policy profile's level.
- You cannot change a priority of an inherited policy profile (see the figure below).



Inheritance of policy profiles

Policy profiles with the same name

If there are two policies with the same names in different hierarchy levels, these policies function according to the following rules:

- Locked settings and the profile activation condition of a higher-level policy profile changes the settings and profile activation condition of a lower-level policy profile (see the figure below).

Child profile inherits settings values from a parent policy profile

- Unlocked settings and the profile activation condition of a higher-level policy profile do not change the settings and profile activation condition of a lower-level policy profile.

How settings are implemented on a managed device

Implementation of effective settings on a managed device can be described as follows:

- The values of all settings that have not been locked are taken from the policy.
- Then they are overwritten with the values of managed application settings.
- And then the locked settings values from the effective policy are applied. Locked settings values change the values of unlocked effective settings.

Managing policies

This section describes managing policies and provides information about viewing the list of policies, creating a policy, modifying a policy, copying a policy, moving a policy, forced synchronization, viewing the policy distribution status chart, and deleting a policy.

Viewing the list of policies

You can view lists of policies created for the Administration Server or for any administration group.

To view a list of policies:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.

2. In the administration group structure, select the administration group for which you want to view the list of policies.

The list of policies appears in tabular format. If there are no policies, the table is empty. You can show or hide the columns of the table, change their order, view only lines that contain a value that you specify, or use search.

Creating a policy

You can create policies; you can also modify and delete existing policies.

To create a policy:

1. In the main menu, go to **Asset management** → **Policies**.

2. Select the administration group for which the policy is to be created:

   - For the root group.

     In this case you can proceed to the next step.

   - For a subgroup:

     a. Click the **Show hierarchy** button at the top of the page.

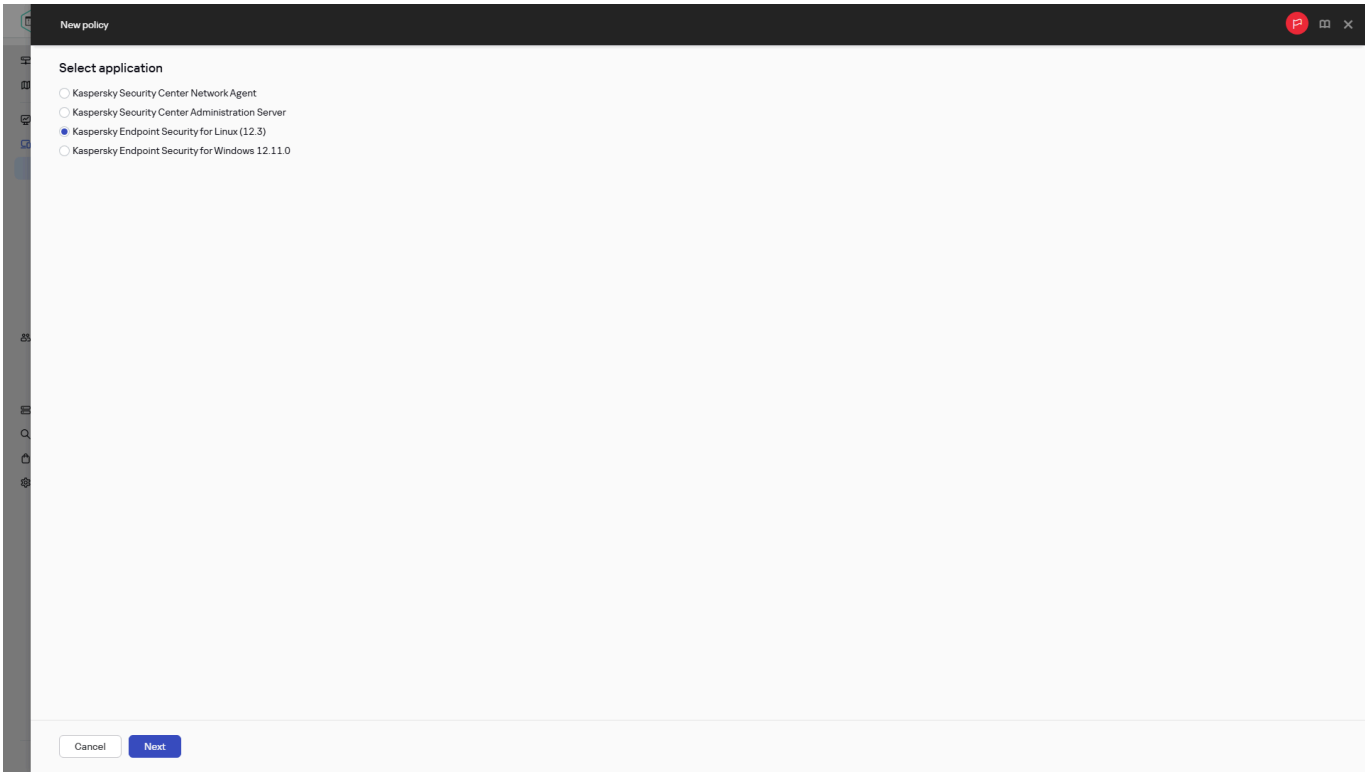
     b. The button name changes to **Hide hierarchy**.

     c. In the window that opens, click the name of the required subgroup.

   The path to the selected group is displayed at the top of the page. If necessary, you can click a link with the administration group name to go to the group. By default, the last link in the path is inactive. Also, you can close the hierarchy window by clicking the **Hide hierarchy** button.

3. Click **Add**.

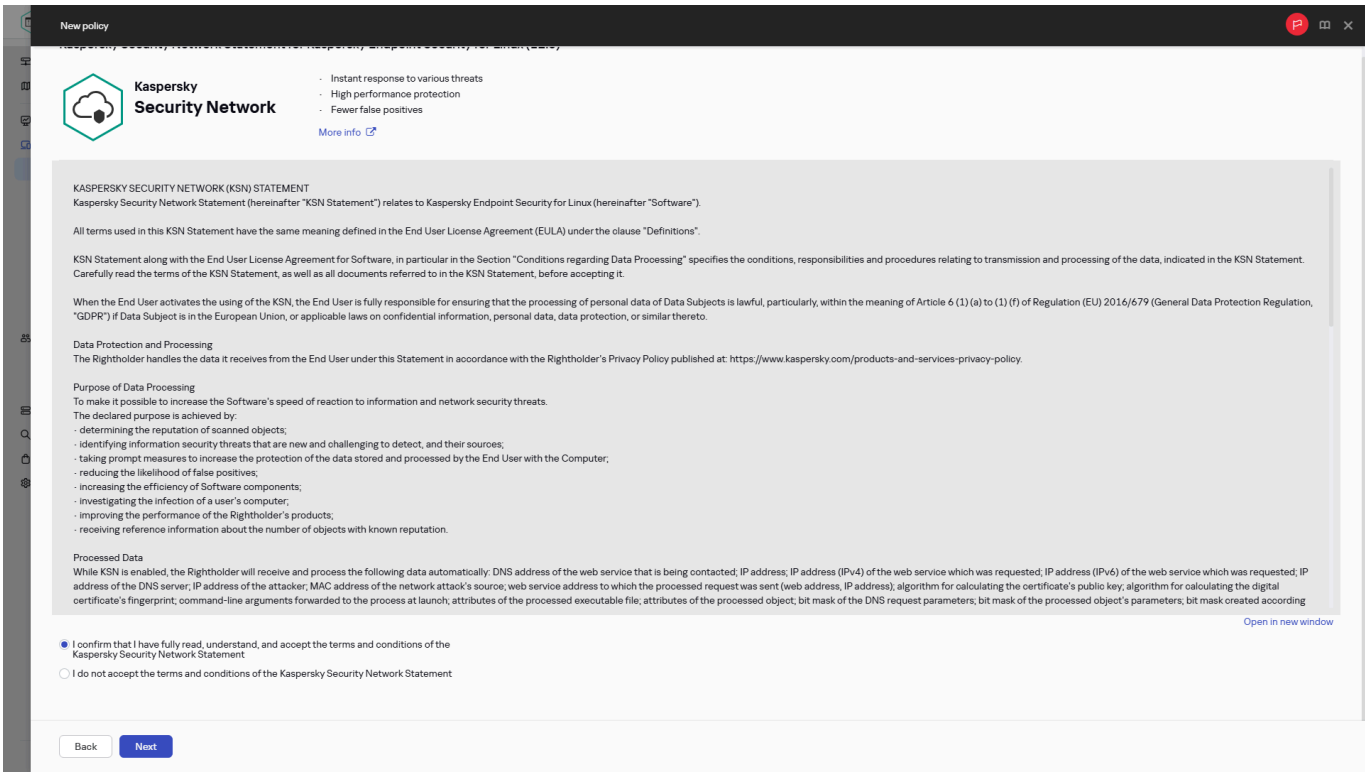
The **Select application** window opens.



Creating a new policy

4. Select the application for which you want to create a policy.

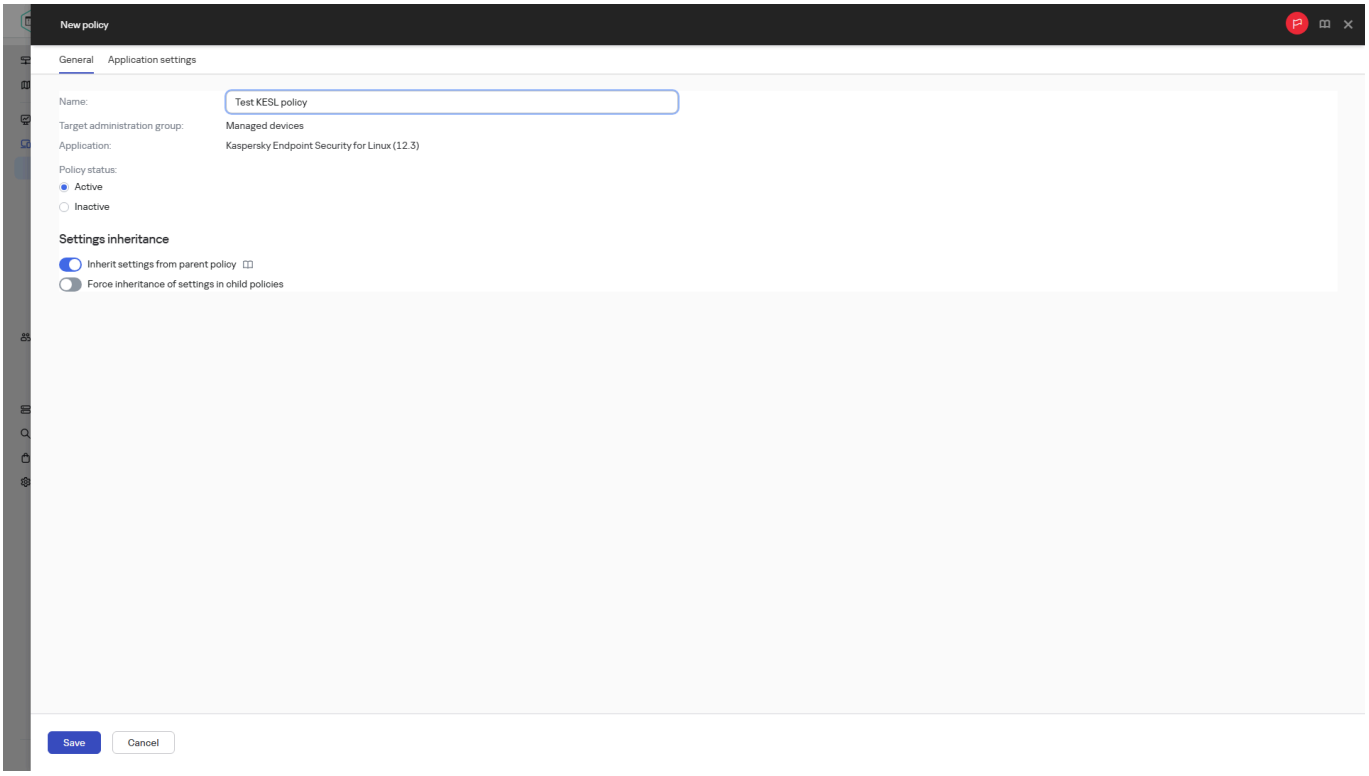
5. If Kaspersky Security Network is supported by the Kaspersky application, for which you are creating a policy, you will be prompted to accept the KSN Statement. If you want to use Kaspersky Security Network, accept the KSN Statement.



Accepting the KSN Statement

6. Click **Next**.

The new policy settings window opens with the **General** tab selected. If you want, change the default name, default status, and default inheritance settings of the policy.



The new policy settings window

7. Select the **Application settings** tab.

   Or, you can click **Save** and exit. The policy will appear in the list of policies, and you can edit its settings later.

8. On the **Application settings** tab, in the left pane select the category that you want and in the results' pane on the right, edit the settings of the policy. You can edit policy settings in each category (section).

   The set of settings depends on the application for which you create a policy. For details, refer to the following:

   - Administration Server configuration
   - Network Agent policy settings
   - Kaspersky Endpoint Security for Linux Help
   - Kaspersky Endpoint Security for Windows Help

   For details about settings of other security applications, refer to the documentation for the corresponding application.

   When editing the settings, you can click **Cancel** to cancel the last operation.

9. Click **Save** to save the policy.

The policy will appear in the list of policies.

General policy settings

General

In the **General** tab, you can modify the policy status and specify the inheritance of policy settings:

- In the **Policy status** block, you can select one of the policy modes:
  - **Active**

    If this option is selected, the policy becomes active.

    By default, this option is selected.

  - **Out-of-office**

    If this option is selected, the policy becomes active when the device leaves the corporate network.

  - **Inactive**

    If this option is selected, the policy becomes inactive, but it is still stored in the **Policies** folder. If required, the policy can be activated.

- In the **Settings inheritance** settings group, you can configure the policy inheritance:
  - **Inherit settings from parent policy**

    If this option is enabled, the policy setting values are inherited from the upper-level group policy and, therefore, are locked.

    By default, this option is enabled.

  - **Force inheritance of settings in child policies**

    If this option is enabled, after policy changes are applied, the following actions will be performed:

    - The values of the policy settings will be propagated to the policies of administration subgroups, that is, to the child policies.
    - In the **Settings inheritance** block of the **General** section in the properties window of each child policy, the **Inherit settings from parent policy** option will be automatically enabled.

    If this option is enabled, the child policies settings are locked.

    By default, this option is disabled.

Event configuration

The **Event configuration** tab allows you to configure event logging and event notification. Events are distributed by importance level on the following tabs:

- **Critical**

  The **Critical** section is not displayed in the Network Agent policy properties.

- **Functional failure**
- **Warning**
- **Info**

In each section, the list shows the types of events and the default event storage term on the Administration Server (in days). Clicking an event type lets you specify the following settings:

- **Event registration**

  You can specify how many days to store the event and select where to store the event:

  - **Export to SIEM system by using Syslog**
  - **Store in the OS event log on device**
  - **Store in the OS event log on Administration Server**

- **Event notifications**

  You can select if you want to be notified about the event in one of the following ways:

  - **Notify by email**
  - **Notify by SMS**
  - **Notify by running an executable file or script**
  - **Notify by SNMP**

  By default, the notification settings specified on the Administration Server properties tab (such as recipient address) are used. If you want, you can change these settings in the **Email**, **SMS**, and **Executable file to be run** tabs.

Also, the **Event configuration** tab displays a notification when new event types are added (for example, in a new version of the application), and enables you to apply the new settings by clicking the **Save** or **Save and close** button.

Revision history

The **Revision history** tab allows you to view the list of the policy revisions and roll back changes made to the policy, if necessary.

Modifying a policy

To modify a policy:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy that you want to modify.

   The policy settings window opens.

3. Specify the general settings and settings of the application for which you create a policy. For details, refer to the following:

   - Administration Server configuration
   - Network Agent policy settings
   - Kaspersky Endpoint Security for Windows documentation

   For details about settings of other security applications, refer to the documentation for that application.

4. Click **Save**.

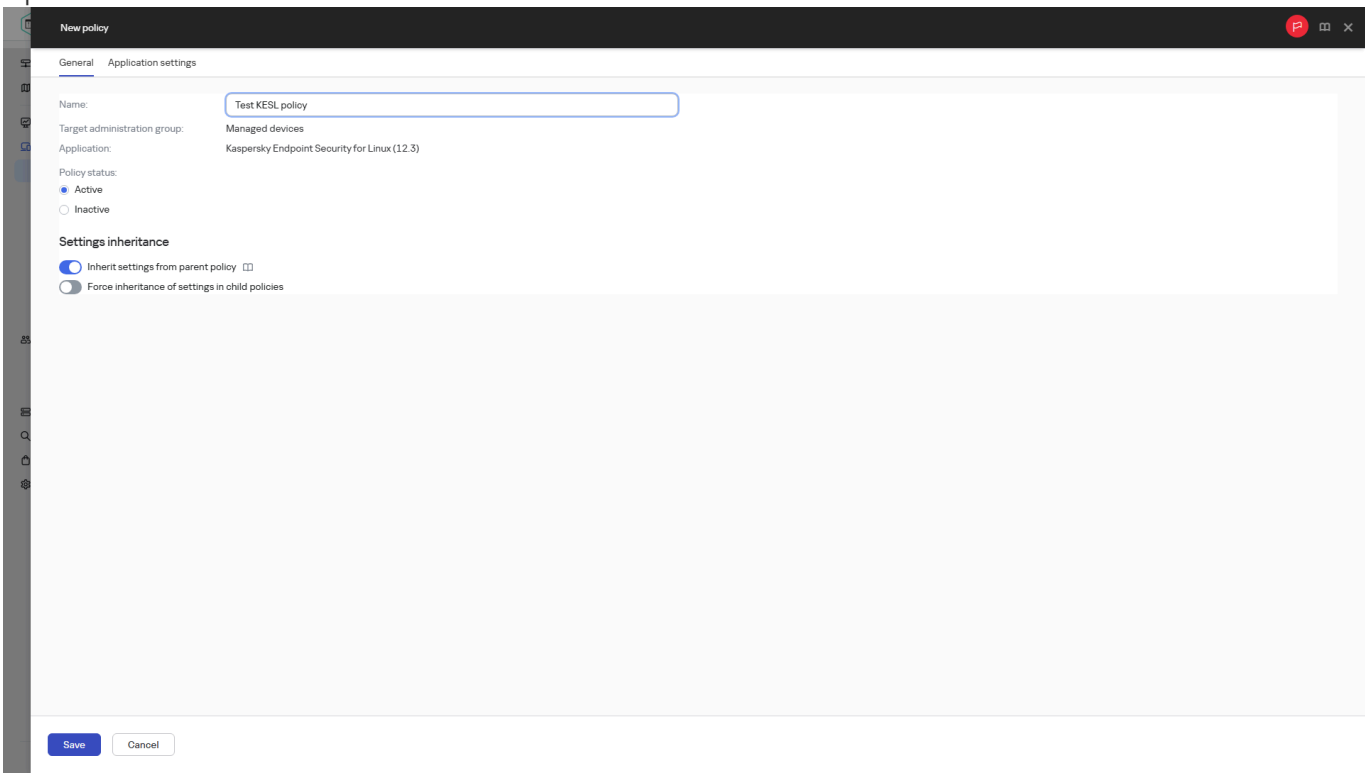
   The changes made to the policy will be saved in the policy properties, and will appear in the **Revision history** section.

Enabling and disabling a policy inheritance option

To enable or disable the inheritance option in a policy:

1. Open the required policy.

2. Open the **General** tab.



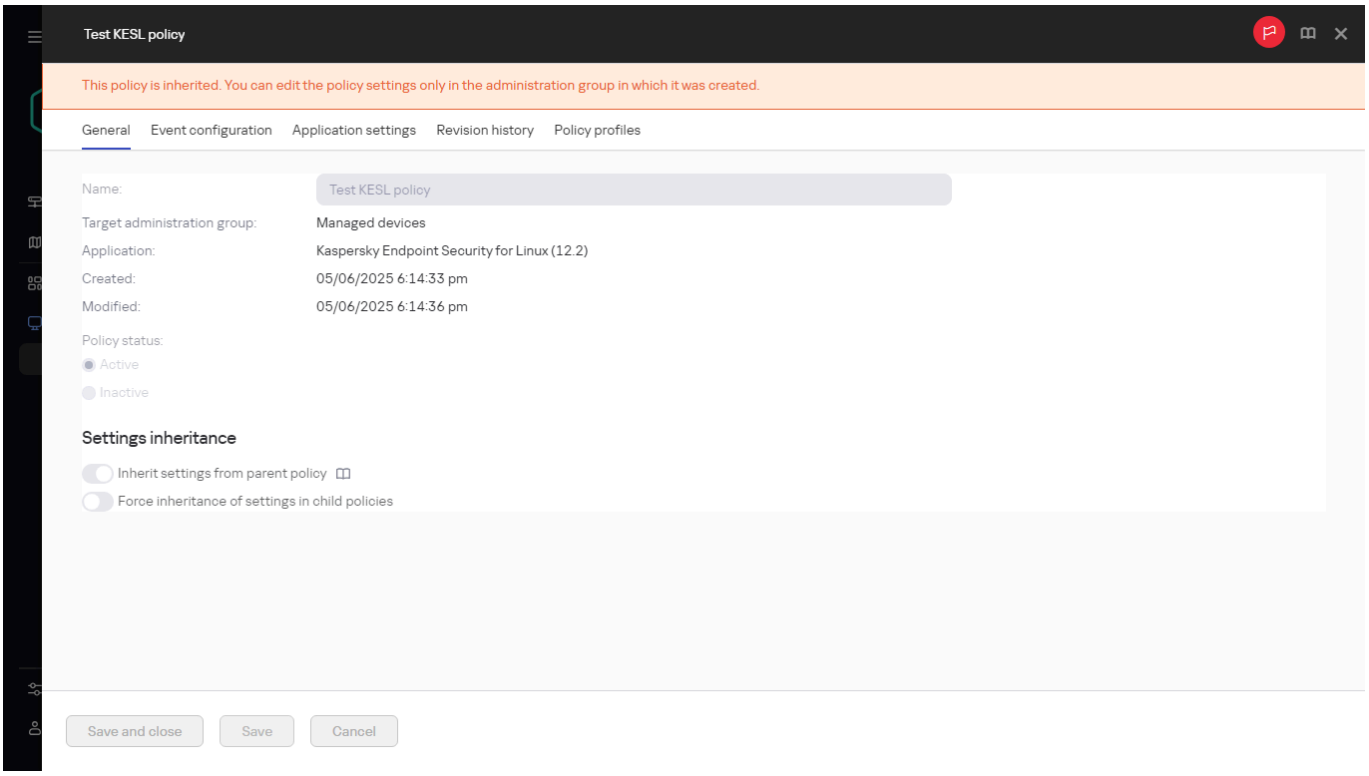
Creating a new policy

3. Enable or disable policy inheritance:

   - If you enable **Inherit settings from parent policy** in a child policy and an administrator locks some settings in the parent policy, then you cannot change these settings in the child policy.

   - If you disable **Inherit settings from parent policy** in a child policy, then you can change all of the settings in the child policy, even if some settings are locked in the parent policy.

   - If you enable **Force inheritance of settings in child policies** in the parent group, this enables the **Inherit settings from parent policy** option for each child policy. In this case, you cannot disable this option for any child policy. All of the settings that are locked in the parent policy are forcibly inherited in the child groups, and you cannot change these settings in the child groups.



Locked policy settings

4. Click the **Save** button to save changes or click the **Cancel** button to reject changes.

By default, the **Inherit settings from parent policy** option is enabled for a new policy.

If a policy has profiles, all of the child policies inherit these profiles.

Copying a policy

You can copy policies from one administration group to another.

To copy a policy to another administration group:

1. In the main menu, go to **Asset management** → **Policies**.

2. Select the check box next to the policy (or policies) that you want to copy.

3. Click the **Copy** button.

   On the right side of the screen, the tree of the administration groups appears.

4. In the tree, select the target group, that is, the group to which you want to copy the policy (or policies).

5. Click the **Copy** button at the bottom of the screen.

6. Click **OK** to confirm the operation.

The policy (policies) will be copied to the target group with all its profiles. The status of each copied policy in the target group will be **Inactive**. You can change the status to **Active** at any time.

If a policy with the name identical to that of the newly moved policy already exists in the target group, the name of the newly moved policy is expanded with the (<next sequence number>) index, for example: (1).

Moving a policy

You can move policies from one administration group to another. For example, you want to delete a group, but you want to use its policies for another group. In this case, you may want move the policy from the old group to the new one before deleting the old group.

> When moving a policy to another administration group, the revision history of the policy is cleared. The **Revision history** tab displays information about the latest change only.

To move a policy to another administration group:

1. In the main menu, go to **Asset management** → **Policies**.

2. Select the check box next to the policy (or policies) that you want to move.

3. Click the **Move** button.

   On the right side of the screen, the tree of the administration groups appears.

4. In the tree, select the target group, that is, the group to which you want to move the policy (or policies).

5. Click the **Move** button at the bottom of the screen.

6. Click **OK** to confirm the operation.

   If a policy is not inherited from the source group, it is moved to the target group with all its profiles. The status of the policy in the target group is **Inactive**. You can change the status to **Active** at any time.

   If a policy is inherited from the source group, it remains in the source group. It is copied to the target group with all its profiles. The status of the policy in the target group is **Inactive**. You can change the status to **Active** at any time.

If a policy with the name identical to that of the newly moved policy already exists in the target group, the name of the newly moved policy is expanded with the (<next sequence number>) index, for example: (1).

Exporting a policy

Open Single Management Platform allows you to save a policy, its settings, and the policy profiles to a KLP file. You can use this KLP file to import the saved policy both to Kaspersky Security Center Windows and Kaspersky Security Center Linux.

To export a policy:

1. In the main menu, go to **Asset management** → **Policies**.
2. Select the check box next to the policy that you want to export.

   You cannot export multiple policies at the same time. If you select more than one policy, the **Export** button will be disabled.
3. Click the **Export** button.
4. In the opened **Save as** window, specify the policy file name and path. Click the **Save** button.

   The **Save as** window is displayed only if you use Google Chrome, Microsoft Edge, or Opera. If you use another browser, the policy file is automatically saved in the **Downloads** folder.

Importing a policy

Open Single Management Platform allows you to import a policy from a KLP file. The KLP file contains the exported policy, its settings, and the policy profiles.

To import a policy:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the **Import** button.
3. Click the **Browse** button to choose a policy file that you want to import.
4. In the opened window, specify the path to the KLP policy file, and then click the **Open** button. Note that you can select only one policy file.

   The policy processing starts.
5. After the policy is processed successfully, select the administration group to which you want to apply the policy.
6. Click the **Complete** button to finish the policy import.

The notification with the import results appears. If the policy is imported successfully, you can click the **Details** link to view the policy properties.

After a successful import, the policy is displayed in the policy list. The settings and profiles of the policy are also imported. Regardless of the policy status that was selected during the export, the imported policy is inactive. You can change the policy status in the policy properties.

> If the newly imported policy has a name identical to that of an existing policy, the name of the imported policy is expanded with the **(<next sequence number>)** index, for example: **(1)**, **(2)**.

Forced synchronization

Although Open Single Management Platform automatically synchronizes the status, settings, tasks, and policies for managed devices, in some cases the administrator must know for certain, at a given moment, whether synchronization has already been performed for a specified device.

Synchronizing a single device

To force synchronization between the Administration Server and a managed device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.
2. Click the name of the device that you want to synchronize with the Administration Server.

   A property window opens with the **General** section selected.
3. Click the **Force synchronization** button.

The application synchronizes the selected device with the Administration Server.

Synchronizing multiple devices

To force synchronization between the Administration Server and multiple managed devices:

1. Open the device list of an administration group or a device selection:
   - Run a device selection to view the device list.
   - In the main menu, go to **Asset management** → **Assets (Devices)**, and then select the administration group that contains devices to synchronize:
     - For the root group, you can proceed to the next step.
     - For a subgroup, click the **Show hierarchy** button at the top of the page, and then in the window that opens, click the name of the subgroup.

       The path to the selected group is displayed at the top of the page. If necessary, you can click a link with the administration group name to go to the group. By default, the last link in the path is inactive. Also, you can close the hierarchy window by clicking the **Hide hierarchy** button.
2. Select the check boxes next to the devices that you want to synchronize with the Administration Server.

3. Above the list of managed devices, click the ellipsis button (**...**), and then click the **Force synchronization** button.

   The application synchronizes the selected devices with the Administration Server.

4. In the device list, check that the time of last connection to the Administration Server has changed, for the selected devices, to the current time. If the time has not changed, update the page content by clicking the **Refresh** button.

The selected devices are synchronized with the Administration Server.

Viewing the time of a policy delivery

After changing a policy for a Kaspersky application on the Administration Server, the administrator can check whether the changed policy has been delivered to a specific managed device. A policy can be delivered during a regular synchronization or a forced synchronization.

To view the date and time that an application policy was delivered to a managed device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

2. Click the name of the device that you want to synchronize with the Administration Server.

   A property window opens with the **General** section selected.

3. Click the **Applications** tab.

4. Select the application for which you want to view the policy synchronization date.

   The application policy window opens with the **General** section selected and the policy delivery date and time displayed.

Viewing the policy distribution status chart

In Open Single Management Platform, you can view the status of policy application on each device in a policy distribution status chart.

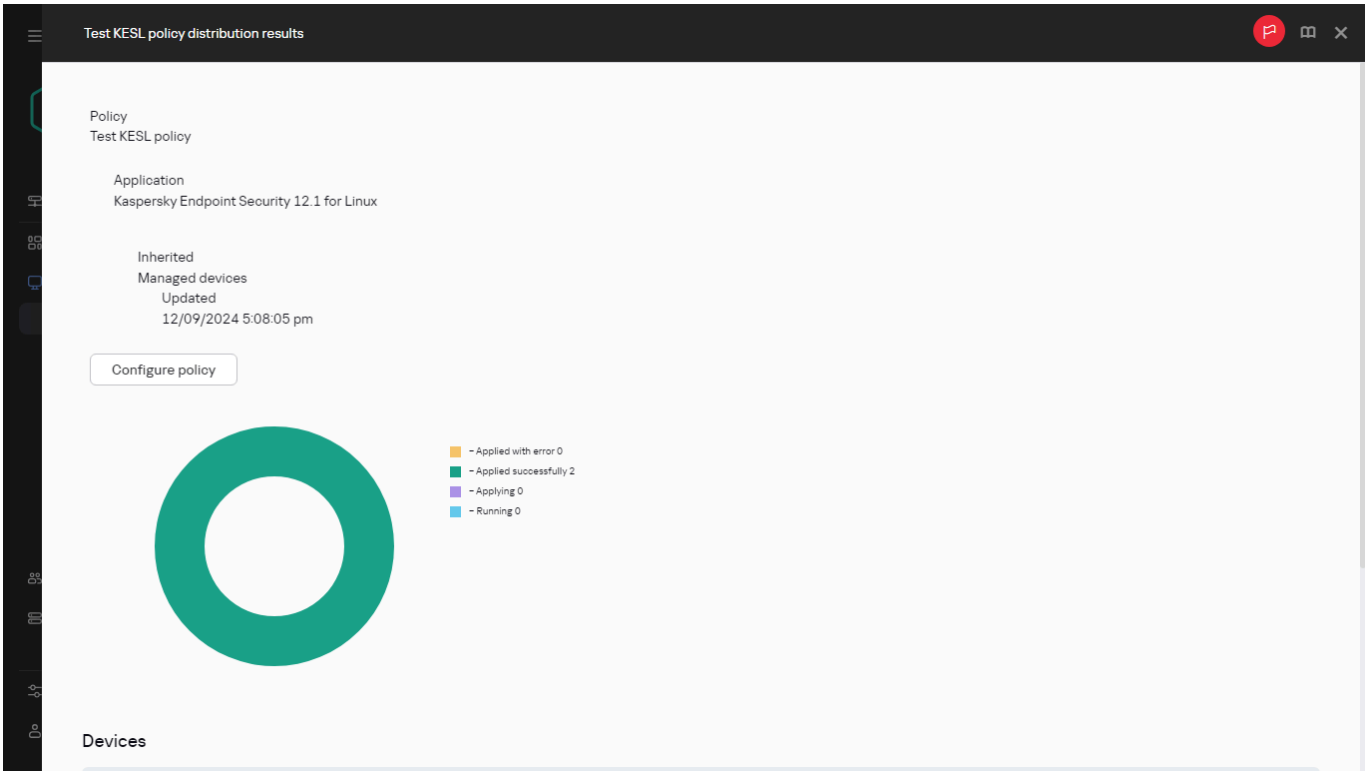
To view the policy distribution status on each device:

1. In the main menu, go to **Asset management** → **Policies**.

2. Select check box next to the name of the policy for which you want to view the distribution status on devices.

3. In the menu that appears, select the **Distribution** link.

The **<Policy name> distribution results** window opens.



Policy distribution status chart

4. In the **<Policy name> distribution results** window that opens, the **Status description** of the policy is displayed.

You can change number of results displayed in the list with policy distribution. The maximum number of devices is 100,000.

To change the number of devices displayed in the list with policy distribution results:

1. In the main menu, go to **Settings** → **Interface options**.
2. In the **Limit of devices displayed in policy distribution results**, enter the number of devices (up to 100,000).

   By default, the number is 5000.
3. Click **Save**.

   The settings are saved and applied.

Activating a policy automatically at the Virus outbreak event

To make a policy perform automatic activation at a Virus outbreak event:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens, with the **General** tab selected.
2. Select the **Virus outbreak** section.
3. In the right pane, click the **Configure policies to activate when a virus outbreak event occurs** link.

   The **Policy activation** window opens.

4. In the section relating to the component that detects a virus outbreak—Anti-Virus for workstations and file servers, Anti-Virus for mail servers, or Anti-Virus for perimeter defense—select the option button next to the entry you want, and then click **Add**.

   A window opens with the **Managed devices** administration group.

5. Click the chevron icon (>) next to **Managed devices**.

   A hierarchy of administration groups and their policies is displayed.

6. In the hierarchy of administration groups and their policies, click the name of a policy or policies that are activated when a virus outbreak is detected.

   To select all policies in the list or in a group, select the check box next to the required name.

7. Click the **Save** button.

   The window with the hierarchy of administration groups and their policies is closed.

The selected policies are added to the list of policies that are activated when a virus outbreak is detected. The selected policies are activated at the virus outbreak, independent whether they are active or inactive.

If a policy has been activated on the Virus outbreak event, you can return to the previous policy only by using the manual mode.

Deleting a policy

You can delete a policy if you do not need it anymore. You can delete only a policy that is not inherited in the specified administration group. If a policy is inherited, you can only delete it in the upper-level group for which it was created.

To delete a policy:

1. In the main menu, go to **Asset management** → **Policies**.

2. Select the check box next to the policy that you want to delete, and click **Delete**.

   The **Delete** button becomes unavailable (dimmed) if you select an inherited policy.

3. Click **OK** to confirm the operation.

The policy is deleted together with all its profiles.

Managing policy profiles

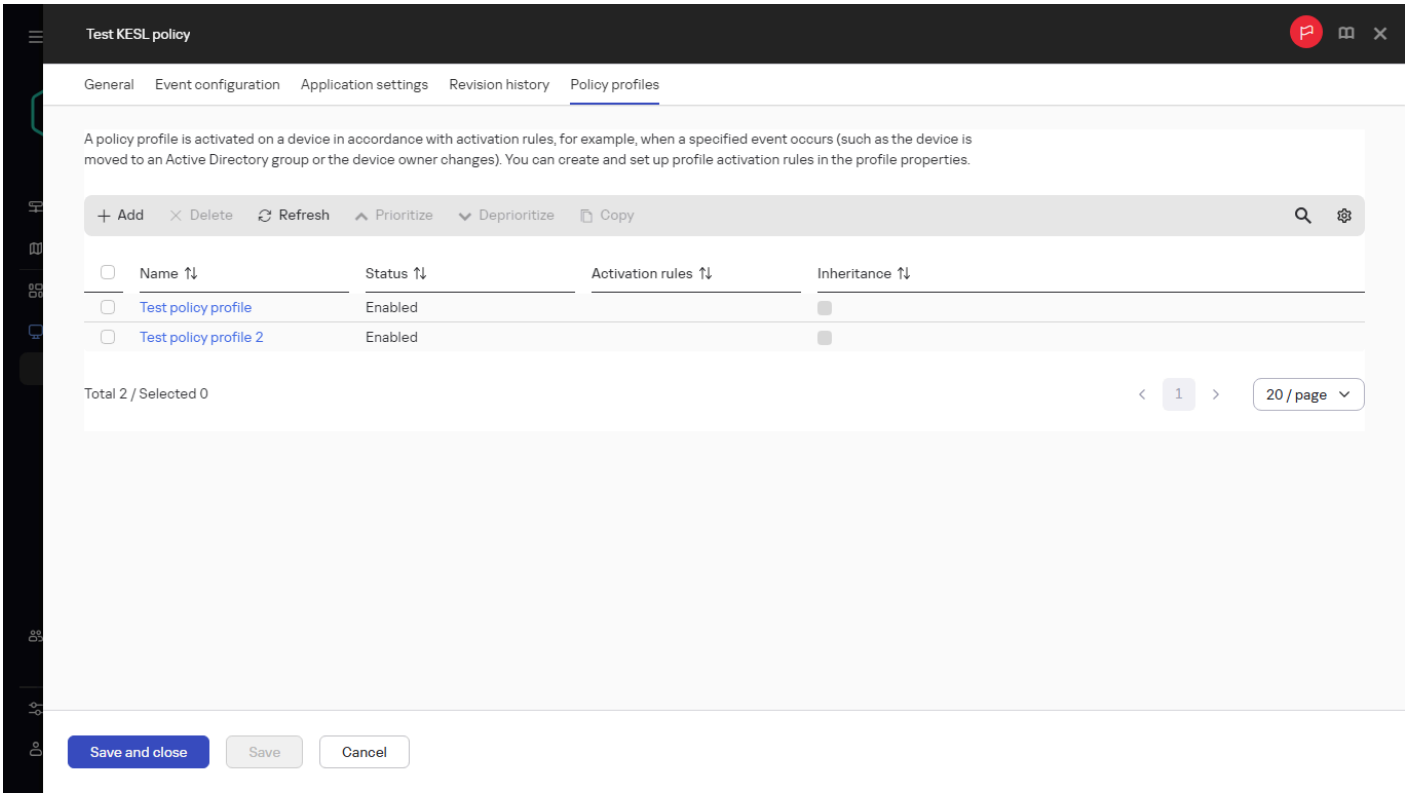
This section describes managing policy profiles and provides information about viewing the profiles of a policy, changing a policy profile priority, creating a policy profile, copying a policy profile, creating a policy profile activation rule, deleting, exporting and importing a policy profile.

Viewing the profiles of a policy

To view profiles of a policy:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the name of the policy whose profiles you want to view.
   The policy properties window opens with the **General** tab selected.
3. Open the **Policy profiles** tab.

The list of policy profiles appears in tabular format. If the policy does not have profiles, an empty table appears.

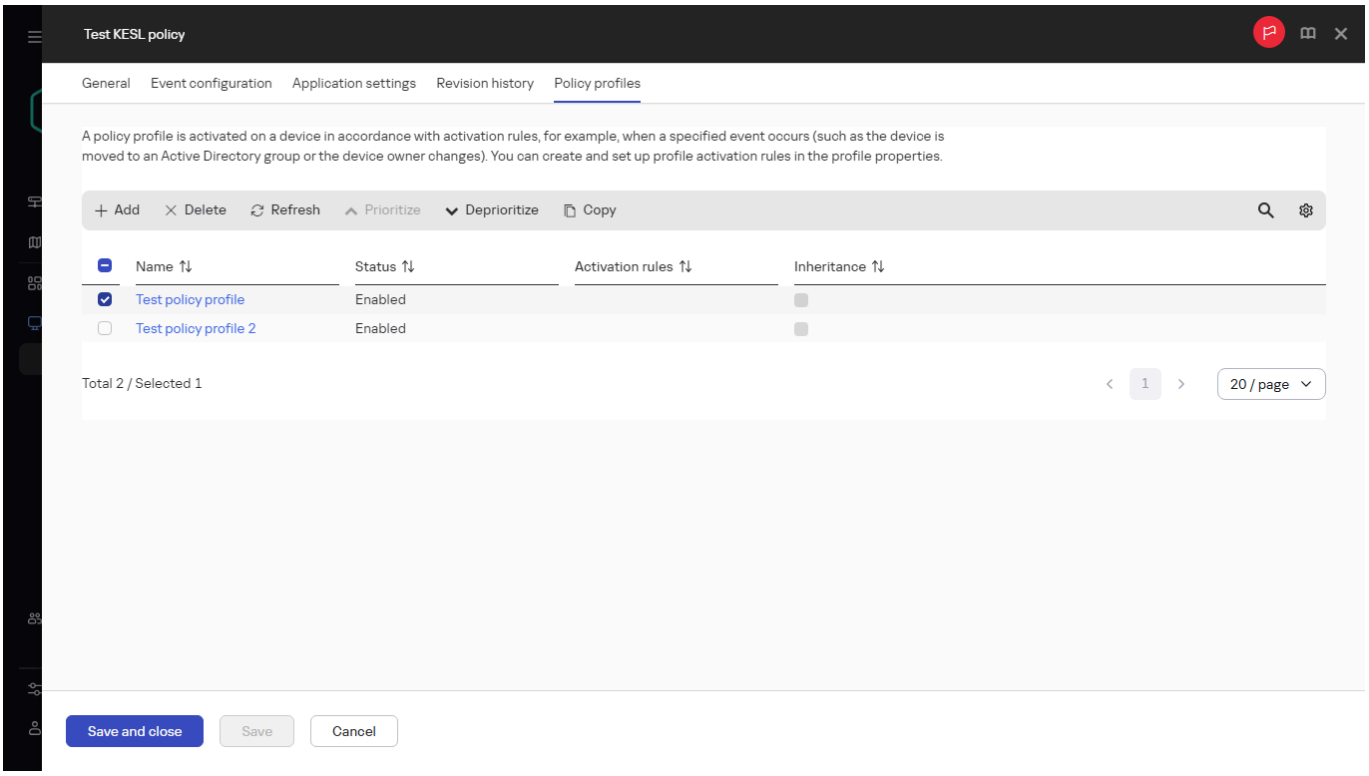


The list of policy profiles

Changing a policy profile priority

To change a policy profile priority:

1. Proceed to the list of profiles of a policy that you want.
   The list of policy profiles appears.
2. On the **Policy profiles** tab, select the check box next to the policy profile for which you want to change priority.
3. Set a new position of the policy profile in the list by clicking **Prioritize** or **Deprioritize**.
   The higher a policy profile is located in the list, the higher its priority.



4. Click the **Save** button.

   Priority of the selected policy profile is changed and applied.

Creating a policy profile

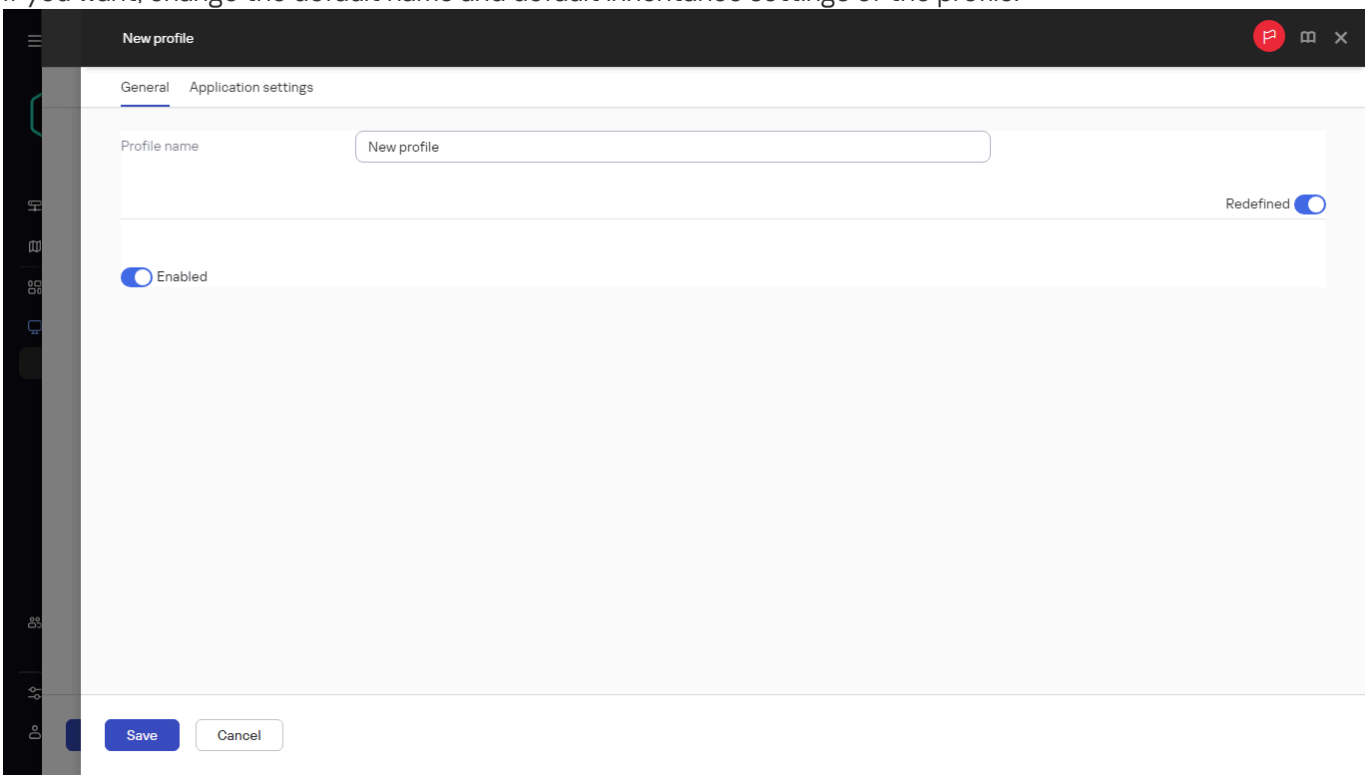
To create a policy profile:

1. Proceed to the list of profiles of the policy that you want.

   The list of policy profiles appears. If the policy does not have profiles, an empty table appears.

2. Click **Add**.

3. If you want, change the default name and default inheritance settings of the profile.



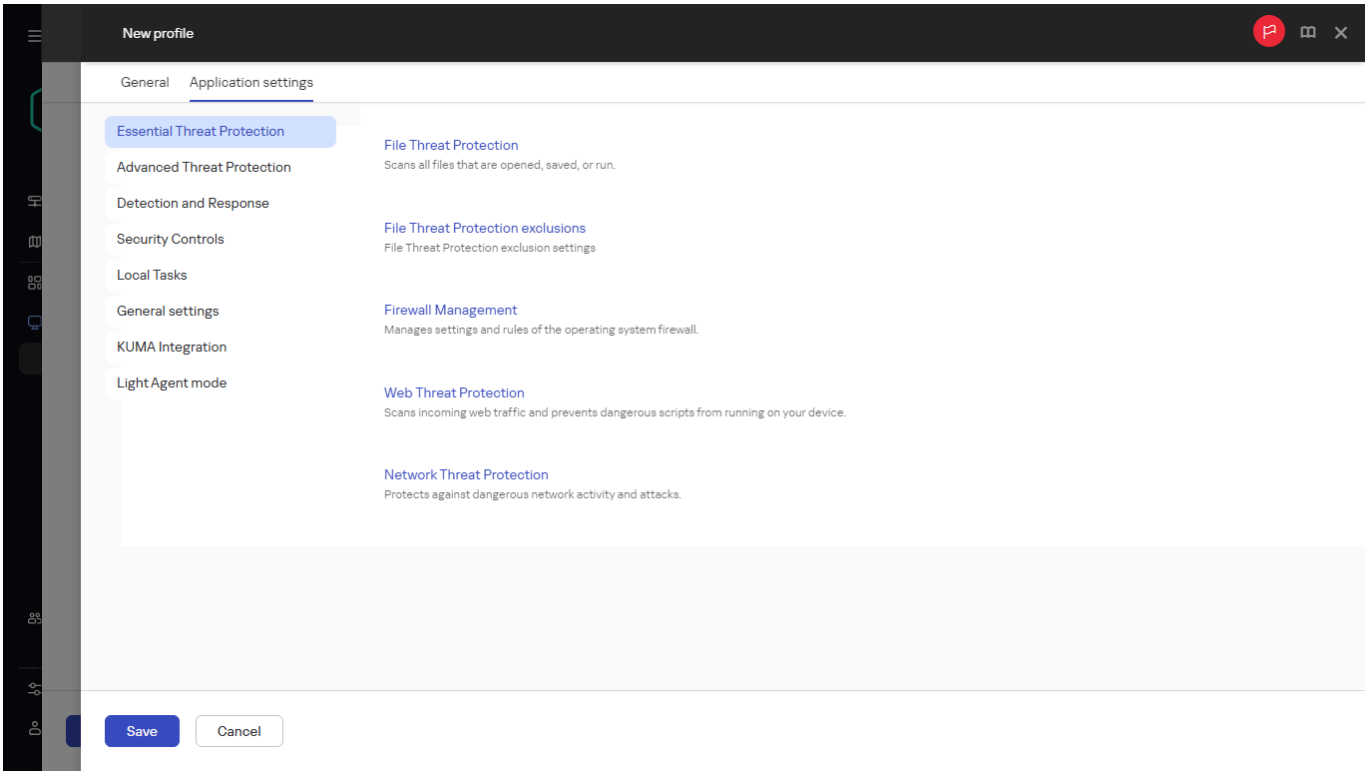
Creating a new policy profile

4. Select the **Application settings** tab.

   Alternatively, you can click **Save** and exit. The profile that you have created appears in the list of policy profiles, and you can edit its settings later.

5. On the **Application settings** tab, in the left pane, select the category that you want and in the results pane on the right, edit the settings for the profile. You can edit policy profile settings in each category (section).

When editing the settings, you can click **Cancel** to cancel the last operation.



Specifying the policy profile settings

6. Click **Save** to save the profile.

   The profile will appear in the list of policy profiles.

Copying a policy profile

You can copy a policy profile to the current policy or to another, for example, if you want to have identical profiles for different policies. You can also use copying if you want to have two or more profiles that differ in only a small number of settings.

To copy a policy profile:

1. Proceed to the list of profiles of a policy that you want.

   The list of policy profiles appears. If the policy does not have profiles, an empty table appears.

2. On the **Policy profiles** tab, select the policy profile that you want to copy.

3. Click **Copy**.

4. In the window that opens, select the policy to which you want to copy the profile.

   You can copy a policy profile to the same policy or to a policy that you specify.

5. Click **Copy**.

   The policy profile is copied to the policy that you selected. The newly copied profile gets the lowest priority. If you copy the profile to the same policy, the name of the newly copied profile will be expanded with the () index, for example: (1), (2).

Later, you can change the settings of the profile, including its name and its priority; the original policy profile will not be changed in this case.

Creating a policy profile activation rule

To create a policy profile activation rule:

1. Proceed to the list of profiles of a policy that you want.

   The list of policy profiles appears.

2. On the **Policy profiles** tab, click the policy profile for which you need to create an activation rule.

   If the list of policy profiles is empty, you can create a policy profile.

3. On the **Activation rules** tab, click the **Add** button.

   The window with policy profile activation rules opens.

4. Specify a name for the rule.

5. Select the check boxes next to the conditions that must affect activation of the policy profile that you are creating:

   - **General rules for policy profile activation**

     Select this check box to set up policy profile activation rules on the device depending on the status of the device offline mode, rule for connection to Administration Server, and tags assigned to the device.

     For this option, specify at the next step:

     - **Device status**

       Defines the condition for device presence on the network:

       - **Online**—The device is on the network, and so the Administration Server is available.
       - **Offline**—The device is on an external network, which means that the Administration Server is not available.
       - **N/A**—The criterion will not be applied.

     - **Network location of the device**

       Choose the condition of policy profile activation (whether the rule is executed or not) and select the rule name.

       The rule defines the network location of the device for connection to the Administration Server, whose conditions must be met (or must not be met) for activation of the policy profile.

       A network location description of devices for connection to an Administration Server can be created or configured in a Network Agent switching rule.

   - **Rules for specific device owner**

For this option, specify at the next step:

- **Device owner**

  Enable this option to configure and enable the rule for profile activation on the device according to its owner. In the drop-down list under the check box, you can select a criterion for the profile activation:

  - The device belongs to the specified owner ("=" sign).
  - The device does not belong to the specified owner ("#" sign).

    If this option is enabled, the profile is activated on the device in accordance with the criterion configured. You can specify the device owner when the option is enabled. If this option is disabled, the profile activation criterion is not applied. By default, this option is disabled.

- **Device owner is included in an internal security group**

  Enable this option to configure and enable the rule of profile activation on the device by the owner's membership in an internal security group of Open Single Management Platform. In the drop-down list under the check box, you can select a criterion for the profile activation:

  - The device owner is a member of the specified security group ("=" sign).
  - The device owner is not a member of the specified security group ("#" sign).

    If this option is enabled, the profile is activated on the device in accordance with the criterion configured. You can specify a security group of Open Single Management Platform. If this option is disabled, the profile activation criterion is not applied. By default, this option is disabled.

- **Rules for hardware specifications**

  Select this check box to set up rules for policy profile activation on the device depending on the memory volume and the number of logical processors.

  For this option, specify at the next step:

  - **RAM size, in MB**

    Enable this option to configure and enable the rule of profile activation on the device by the RAM volume available on that device. In the drop-down list under the check box, you can select a criterion for the profile activation:

    - The device RAM size is less than the specified value ("<" sign).
    - The device RAM size is greater than the specified value (">" sign).

    If this option is enabled, the profile is activated on the device in accordance with the criterion configured. You can specify the RAM volume on the device. If this option is disabled, the profile activation criterion is not applied. By default, this option is disabled.

- **Number of logical processors**

  Enable this option to configure and enable the rule of profile activation on the device by the number of logical processors on that device. In the drop-down list under the check box, you can select a criterion for the profile activation:

  - The number of logical processors on the device is less than or equal to the specified value ("<" sign).
  - The number of logical processors on the device is greater than or equal to the specified value (">" sign).

  If this option is enabled, the profile is activated on the device in accordance with the criterion configured. You can specify the number of logical processors on the device. If this option is disabled, the profile activation criterion is not applied. By default, this option is disabled.

- **Rules for role assignment**

  For this option, specify at the next step:

  - **Activate policy profile by specific role of device owner**

    Select this option to configure and enable the rule of profile activation on the device depending on the owner's role. Add the role manually from the list of existing roles.

    If this option is enabled, the profile is activated on the device in accordance with the criterion configured.

- **Rules for tag usage**

  Select this check box to set up rules for policy profile activation on the device depending on the tags assigned to the device. You can activate the policy profile to the devices that either have the selected tags or do not have them.

  For this option, specify at the next step:

  - **Tag list**

    In the list of tags, specify the rule for device inclusion in the policy profile by selecting the check boxes next to the relevant tags.

    You can add new tags to the list by entering them in the field over the list and clicking the **Add** button.

    The policy profile includes devices with descriptions containing all the selected tags. If check boxes are cleared, the criterion is not applied. By default, these check boxes are cleared.

  - **Apply to devices without the specified tags**

    Enable this option if you have to invert your selection of tags.

    If this option is enabled, the policy profile includes devices with descriptions that contain none of the selected tags. If this option is disabled, the criterion is not applied.

    By default, this option is disabled.

- **Rules for Active Directory usage**

  Select this check box to set up rules for policy profile activation on the device depending on the presence of the device in an Active Directory organizational unit (OU), or on membership of the device (or its owner) in an Active Directory security group.

  For this option, specify at the next step:

  - **Device owner's membership in an Active Directory security group**

    If this option is enabled, the policy profile is activated on the device whose owner is a member of the specified security group. If this option is disabled, the profile activation criterion is not applied. By default, this option is disabled.

  - **Device membership in Active Directory security group**

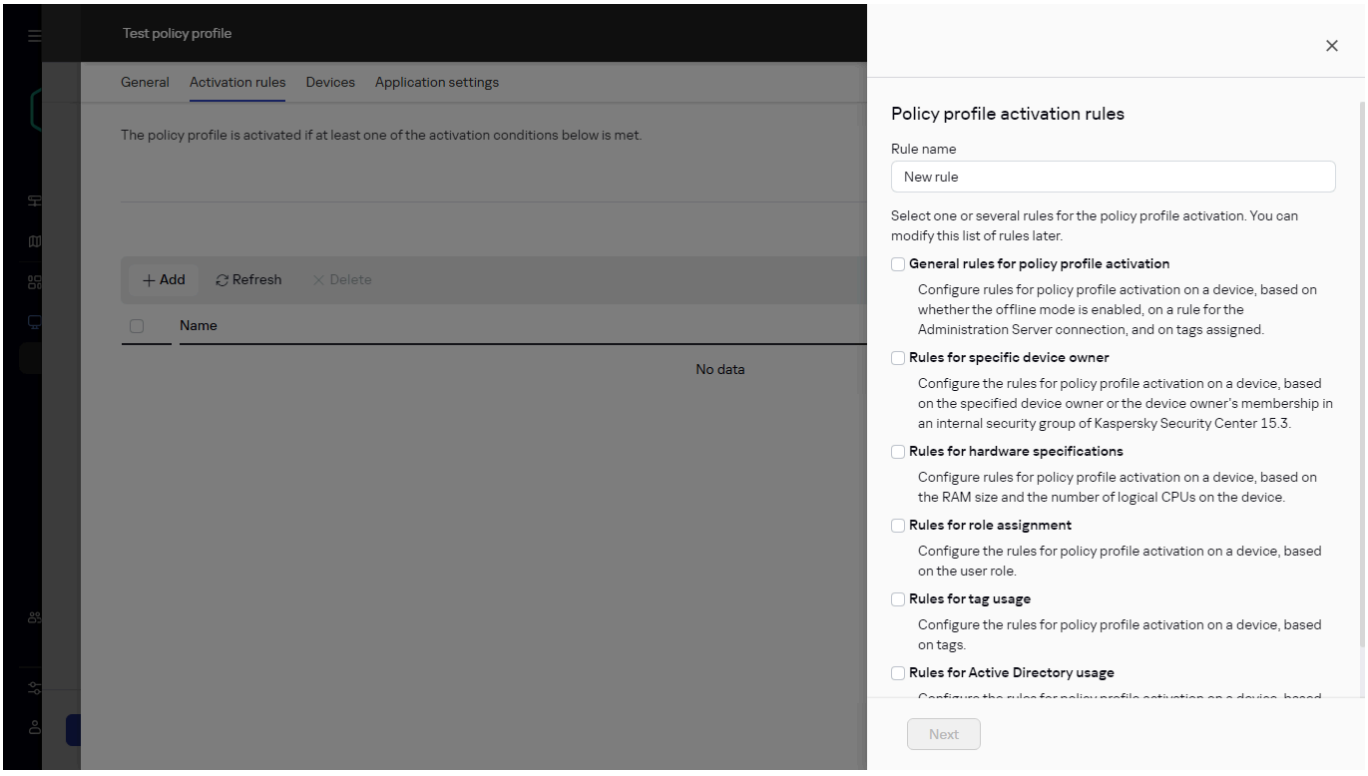
    If this option is enabled, the policy profile is activated on the device. If this option is disabled, the profile activation criterion is not applied. By default, this option is disabled.

  - **Device allocation in Active Directory organizational unit**

    If this option is enabled, the policy profile is activated on the device which is included in the specified Active Directory organizational unit (OU). If this option is disabled, the profile activation criterion is not applied.

    By default, this option is disabled.

The number of additional pages of the wizard depends on the settings that you select at the first step. You can modify policy profile activation rules later.



Creating a policy profile activation rule

6. Check the list of the configured parameters. If the list is correct, click **Create**.

   The profile will be saved. The profile will be activated on the device when activation rules are triggered.

Policy profile activation rules created for the profile are displayed in the policy profile properties on the **Activation rules** tab. You can modify or remove any policy profile activation rule.

> Multiple activation rules can be triggered simultaneously.

Deleting a policy profile

To delete a policy profile:

1. Proceed to the list of profiles of a policy that you want.

   The list of policy profiles appears.

2. On the **Policy profiles** tab, select the check box next to the policy profile that you want to delete, and click **Delete**.

3. In the window that opens, click **Delete** again.

   The policy profile is deleted. If the policy is inherited by a lower-level group, the profile remains in that group, but becomes the policy profile of that group. This is done to eliminate significant change in settings of the managed applications installed on the devices of lower-level groups.

Exporting and importing a policy profile

You can save a policy profile to a KLPROF file without exporting a policy, and then use this KLPROF file to import the saved profile to the policy.

This allows you to restore the policy profile settings quickly, instead of configuring them again.

A policy profile is imported to the policy of the same version of the application from which you exported the profile. For applications that support backward compatibility of policies (for example, Kaspersky Endpoint Security for Windows), you can import policy profiles to different versions of the policy.

To export a policy profile:

1. Proceed to the list of profiles of a policy that you want.

   The list of policy profiles appears.

2. On the **Policy profiles** tab, click the policy profile that you want to export.

3. In the toolbar, click the **Export** button.

4. In the window that opens, specify the directory to which you want to save the file with the policy profile.

   You can change the file name, if necessary.

5. Click the **Save** button.

If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

To import a policy profile:

1. Proceed to the list of profiles of a policy that you want.

   The list of policy profiles appears.

2. In the toolbar, click the **Import** button.

3. In the window that opens, select the KLPROF file with the required policy profile, and then click the **Open** button.

   The maximum size of the imported file is 4 MB.

If the operation is completed successfully, an appropriate message is displayed on the screen. Otherwise, an error message is displayed.

The policy profile is imported with the name it had when you exported it, not with the name of the KLPROF file. The file is imported with the **Disabled** status and the lowest priority, regardless of the priority the policy profile had before the export.

You cannot perform the import:

- If the policy already has a profile with the same name.
- If the number of profiles in the policy is 100.

Network Agent policy settings

To configure the Network Agent policy:

1. In the main menu, go to **Asset management** → **Policies**.

2. Click the name of the Network Agent policy.

   The properties window of the Network Agent policy opens. The properties window contains the tabs and settings described below.

See the comparison table detailing how the settings below apply, depending on the type of operating system used.

General

On this tab, you can modify the policy name, policy status, and specify the inheritance of policy settings:

- In the **Name** field, you can modify the policy name.
- In the **Policy status** block, you can select one of the following policy modes:
  - **Active**

    If this option is selected, the policy becomes active.

    By default, this option is selected.

  - **Inactive**

    If this option is selected, the policy becomes inactive, but it is still stored in the **Policies** folder. If required, the policy can be activated.

- In the **Settings inheritance** settings group, you can configure the policy inheritance:
  - **Inherit settings from parent policy**

    If this option is enabled, the policy setting values are inherited from the upper-level group policy and, therefore, are locked.

    By default, this option is enabled.

  - **Force inheritance of settings in child policies**

    If this option is enabled, after policy changes are applied, the following actions will be performed:

    - The values of the policy settings will be propagated to the policies of administration subgroups, that is, to the child policies.
    - In the **Settings inheritance** block of the **General** section in the properties window of each child policy, the **Inherit settings from parent policy** option will be automatically enabled.

    If this option is enabled, the child policies settings are locked.

    By default, this option is disabled.

Event configuration

On this tab, you can configure event logging and event notification. Events are distributed according to importance level in the following sections:

- **Functional failure**
- **Warning**
- **Info**

In each section, the list shows the types of events and the default event storage period on the Administration Server (in days). After you click the event type, you can specify the settings of event logging and notifications about events selected in the list. By default, common notification settings specified for the entire Administration Server are used for all event types. However, you can change specific settings for the required event types.

For example, in the **Warning** section, you can configure the **Security issue has occurred** event type. Such events may happen, for instance, when the free disk space of a distribution point is less than 2 GB (at least 4 GB are required to install applications and download updates remotely). To configure the **Security issue has occurred** event, click it and specify where to store the occurred events and how to notify about them.

> If the Network Agent detects a security issue, you can manage this issue by using the settings of a managed device.

Application settings

Settings

In the **Settings** section, you can configure the Network Agent policy:

- **Distribute files through distribution points only**

  If this option is enabled, Network Agents on managed devices retrieve updates from distribution points only.

  If this option is disabled, Network Agents on managed devices retrieve updates from distribution points or from Administration Server.

  > Note that the security applications on managed devices retrieve updates from the source set in the update task for each security application. If you enable the **Distribute files through distribution points only** option, make sure that Open Single Management Platform is set as an update source in the update tasks.

  By default, this option is disabled.

- **Maximum size of event queue, in MB**

  In this field you can specify the maximum space on the drive that an event queue can occupy.

  The default value is 2 megabytes (MB).

- **Application is allowed to retrieve policy's extended data on device**

  Network Agent installed on a managed device transfers information about the applied security application policy to the security application (for example, Kaspersky Endpoint Security for Linux). You can view the transferred information in the security application interface.

Network Agent transfers the following information:

- Time of the policy delivery to the managed device.
- Name of the active policy at the moment of the policy delivery to the managed device.
- Name of the out-of-office policy at the moment of the policy delivery to the managed device (not available for the Network Agent for Linux).
- Name and full path to the administration group that contained the managed device at the moment of the policy delivery to the managed device.
- List of active policy profiles with their names and priorities at the moment of the policy delivery to the managed device.

You can use the information to ensure the correct policy is applied to the device and for troubleshooting purposes. By default, this option is disabled.

- **Protect the Network Agent service against unauthorized removal or termination, and prevent changes to the settings**

  When this option is enabled, after Network Agent is installed on a managed device, the component cannot be removed or reconfigured without required privileges. The Network Agent service cannot be stopped. This option has no effect on domain controllers.

  Enable this option to protect Network Agent on workstations operated with local administrator rights.

  By default, this option is disabled.

- **Use uninstallation password**

  If this option is enabled, by clicking the **Modify** button you can specify the password for the klmover utility.

  By default, this option is disabled.

  Disable this option to uninstall Network Agent remotely.

Repositories

In the **Repositories** section, you can select the types of objects whose details will be sent from Network Agent to Administration Server:

- **Details of installed applications**

  If this option is enabled, information about applications installed on client devices is sent to the Administration Server.

  By default, this option is enabled.

- **Include information about patches**

  Information about patches of applications installed on client devices is sent to the Administration Server. Enabling this option may increase the load on the Administration Server and DBMS, as well as cause increased volume of the database.

By default, this option is enabled. It is available only for Windows.

- **Details of Windows Update updates**

  If this option is enabled, information about mandatory Microsoft Windows Update updates that must be installed on client devices is sent to the Administration Server.

  By default, this option is enabled. It is available only for Windows.

  Information about optional Microsoft Windows updates is not being sent to the Administration Server.

- **Details of software vulnerabilities and corresponding updates**

  If this option is enabled, information about vulnerabilities in third-party software (including Microsoft software), detected on managed devices, and about software updates to fix third-party vulnerabilities (not including Microsoft software) is sent to the Administration Server.

  Selecting this option (**Details of software vulnerabilities and corresponding updates**) increases the network load, Administration Server disk load, and Network Agent resource consumption.

  By default, this option is enabled. It is available only for Windows.

  To manage software updates of Microsoft software, use the **Details of Windows Update updates** option.

- **Hardware registry details**

  Network Agent installed on a device sends information about the device hardware to the Administration Server. You can view the hardware details in the device properties.

  Ensure that the lshw utility is installed on Linux devices from which you want to fetch hardware details. Hardware details fetched from virtual machines may be incomplete depending on the hypervisor used.

If modification of some settings in this section is prohibited by the Network Agent policy, you cannot modify these settings.

Software updates and vulnerabilities

In the **Software updates and vulnerabilities** section, you can enable scanning of executable files for vulnerabilities:

- **Scan executable files for vulnerabilities when running them**

  If this option is enabled, executable files are scanned for vulnerabilities when they are run.

  By default, this option is enabled.

Restart management

In the **Restart management** section, you can specify the action to be performed if the operating system of a managed device has to be restarted for correct use, installation, or uninstallation of an application:

- **Do not restart the operating system**

  Client devices are not restarted automatically after the operation. To complete the operation, you must restart a device (for example, manually or through a device management task). Information about the required restart is saved in the task results and in the device status. This option is suitable for tasks on servers and other devices where continuous operation is critical.

- **Restart the operating system automatically, if necessary**

  Client devices are always restarted automatically if a restart is required for completion of the operation. This option is useful for tasks on devices that provide for regular pauses in their operation (shutdown or restart).

- **Prompt user for action**

  The restart reminder is displayed on the screen of the client device, prompting the user to restart it manually. Some advanced settings can be defined for this option: text of the message for the user, the message display frequency, and the time interval after which a restart will be forced (without the user's confirmation). This option is most suitable for workstations where users must be able to select the most convenient time for a restart.

  By default, this option is selected.

  - **Repeat the prompt every (min)**

    If this option is enabled, the application prompts the user to restart the operating system with the specified frequency.

    By default, this option is enabled. The default interval is 5 minutes. Available values are between 1 and 1440 minutes.

    If this option is disabled, the prompt is displayed only once.

  - **Force restart after (min)**

    After prompting the user, the application forces restart of the operating system upon expiration of the specified time interval.

    By default, this option is enabled. The default delay is 30 minutes. Available values are between 1 and 1440 minutes.

  - **Force closure of applications in blocked sessions**

    Running applications may prevent a restart of the client device. For example, if a document is being edited in a word processing application and is not saved, the application does not allow the device to restart.

    If this option is enabled, such applications on a locked device are forced to close before the device restart. As a result, users may lose their unsaved changes.

If this option is disabled, a locked device is not restarted. The task status on this device states that a device restart is required. Users have to manually close all applications running on locked devices and restart these devices.

By default, this option is disabled.

Manage patches and updates

In the Manage patches and updates section, you can configure the download and distribution of updates, as well as the installation of patches, on managed devices:

- **Automatically install applicable updates and patches for components that have the Undefined status**

  If this option is enabled, Kaspersky patches that have the *Undefined* approval status are automatically installed on managed devices immediately after they are downloaded from update servers.

  If this option is disabled, Kaspersky patches that have been downloaded and tagged with the *Undefined* status will be installed only after you change their status to *Approved*.

  By default, this option is enabled.

- **Download updates and anti-virus databases from Administration Server in advance (recommended)**

  If this option is enabled, the offline model of update download is used. When the Administration Server receives updates, it notifies Network Agent (on devices where it is installed) of the updates that will be required for managed applications. When Network Agent receives information about these updates, it downloads the relevant files from the Administration Server in advance. At the first connection with Network Agent, the Administration Server initiates an update download. After Network Agent downloads all the updates to a client device, the updates become available for applications on that device.

  When a managed application on a client device attempts to access Network Agent for updates, Network Agent checks whether it has all required updates. If the updates are received from the Administration Server not more than 25 hours before they were requested by the managed application, Network Agent does not connect to the Administration Server but supplies the managed application with updates from the local cache instead. Connection with the Administration Server may not be established when Network Agent provides updates to applications on client devices, but connection is not required for updating.

  If this option is disabled, the offline model of update download is not used. Updates are distributed according to the schedule of the update download task.

  By default, this option is enabled.

Connectivity

The **Connectivity** section includes three subsections:

- **Network**
- **Connection profiles**
- **Connection schedule**

In the **Network** subsection, you can configure the connection to Administration Server, enable the use of a UDP port, and specify the UDP port number.

- In the **Connect to Administration Server** settings group, you can configure connection to the Administration Server and specify the time interval for synchronization between client devices and the Administration Server:

  - **Synchronization interval (min)**

    Network Agent synchronizes the managed device with the Administration Server. We recommend that you set the synchronization interval (also referred to as the heartbeat) to 15 minutes per 10,000 managed devices.

    If the synchronization interval is set to less than 15 minutes, synchronization is performed every 15 minutes. If synchronization interval is set to 15 minutes or more, synchronization is performed at the specified synchronization interval.

  - **Compress network traffic**

    If this option is enabled, the speed of data transfer by Network Agent is increased by means of a decrease in the amount of information being transferred and a consequent decreased load on the Administration Server.

    The workload on the CPU of the client computer may increase.

    By default, this check box is enabled.

  - **Open Network Agent ports in Microsoft Windows Firewall**

    If this option is enabled, the ports, necessary for the work of Network Agent, are added to the Microsoft Windows Firewall exclusion list.

    By default, this option is enabled.

  - **Use SSL connection**

    If this option is enabled, connection to the Administration Server is established through a secure port via SSL.

    By default, this option is enabled.

  - **Use the connection gateway on a distribution point (if available), under the default connection settings**

    If this option is enabled, the connection gateway on the distribution point is used under the settings specified in the administration group properties.

    By default, this option is enabled.

- **Use UDP port**

  If you need Network Agent to connect to Administration Server through a UDP port, enable the **Use UDP port** option and specify a **UDP port number**. By default, this option is enabled. The default UDP port to connect to Administration Server is 15000.

- **UDP port number**

  In this field you can enter the UDP port number. The default port number is 15000.

  The decimal system is used for records.

- **Use distribution point to force connection to Administration Server**

  Select this option if you selected the **Use this distribution point as a push server** option in the distribution point settings window. Otherwise, the distribution point will not act as a push server.

In the **Connection profiles** subsection, you can specify the network location settings and enable out-of-office mode when Administration Server is not available:

- **Network location settings**

  Network location settings define the characteristics of the network to which the client device is connected and specify rules for Network Agent switching from one Administration Server connection profile to another when those network characteristics are altered.

- **Administration Server connection profiles**

  You can view and add profiles for Network Agent connection to the Administration Server. In this section, you can also create rules for switching Network Agent to different Administration Servers when the following events occur:

  - When the client device connects to a different local network
  - When the device loses connection with the local network of the organization
  - When the connection gateway address is changed or the DNS server address is modified

- **Enable out-of-office mode when Administration Server is not available**

  If this option is enabled, in case of connection through this profile, applications installed on the client device use policy profiles for devices in out-of-office mode, as well as out-of-office policies. If no out-of-office policy has been defined for the application, the active policy will be used.

  If this option is disabled, applications will use active policies.

  By default, this option is disabled.

In the **Connection schedule** subsection, you can specify the time intervals during which Network Agent sends data to the Administration Server:

- **Connect when necessary**

  If this option is selected, the connection is established when Network Agent has to send data to the Administration Server.

  By default, this option is selected.

- **Connect at specified time intervals**

  If this option is selected, Network Agent connects to the Administration Server at a specified time. You can add several connection time periods.

Network polling by distribution points

In the **Network polling by distribution points** section, you can configure automatic polling of the network. You can use the following options to enable the polling and set its frequency:

- **IP ranges**

  If the option is enabled, the distribution point automatically polls IP ranges according to the schedule that you configured by clicking the **Set polling schedule** button.

  If this option is disabled, the distribution point does not poll IP ranges.

  The frequency of IP range polling for Network Agent versions prior to 10.2 can be configured in the **Poll interval (min)** field. The field is available if the option is enabled.

  By default, this option is disabled.

- **Zeroconf**

  If this option is enabled, the distribution point automatically polls the network with IPv6 devices by using zero-configuration networking (also referred to as *Zeroconf*). In this case, the enabled IP range polling is ignored, because the distribution point polls the whole network.

  To start to use Zeroconf, the following conditions must be fulfilled:

  - The distribution point must run Linux.
  - You must install the avahi-browse utility on the distribution point.

  If this option is disabled, the distribution point does not poll networks with IPv6 devices.

  By default, this option is disabled.

- **Domain controllers**

  If the option is enabled, the distribution point automatically polls domain controllers according to the schedule that you configured by clicking the **Set polling schedule** button.

  If this option is disabled, the distribution point does not poll domain controllers.

The frequency of domain controller polling for Network Agent versions prior to 10.2 can be configured in the **Poll interval (min)** field. The field is available if this option is enabled.

By default, this option is disabled.

Network settings for distribution points

In the **Network settings for distribution points** section, you can specify the internet access settings:

- **Use proxy server**
- **Address**
- **Port number**
- **Bypass proxy server for local addresses**

  If this option is enabled, no proxy server is used to connect to devices on the local network.

  By default, this option is disabled.

- **Proxy server authentication**

  If this check box is selected, in the entry fields you can specify the credentials for proxy server authentication.

  By default, this check box is cleared.

KSN Proxy (distribution points)

In the **KSN Proxy (distribution points)** section, you can configure the application to use the distribution point to forward Kaspersky Security Network (KSN) requests from the managed devices:

- **Enable KSN Proxy on the distribution point side**

  The KSN proxy service is run on the device that is used as a distribution point. Use this feature to redistribute and optimize traffic on the network.

  The distribution point sends the KSN statistics, which are listed in the Kaspersky Security Network statement, to Kaspersky.

  By default, this option is disabled. Enabling this option takes effect only if the **Use Administration Server as a proxy server** and **I agree to use Kaspersky Security Network** options are enabled in the Administration Server properties window.

  You can assign a node of an active-passive cluster to a distribution point and enable KSN proxy server on this node.

- **Forward KSN requests to Administration Server**

  The distribution point forwards KSN requests from the managed devices to the Administration Server.

  By default, this option is enabled.

- **Access KSN Cloud/KPSN directly over the internet**

  The distribution point forwards KSN requests from managed devices to the KSN Cloud or KPSN. The KSN requests generated on the distribution point itself are also sent directly to the KSN Cloud or KPSN.

- **TCP port**

  The number of the TCP port that the managed devices will use to connect to KSN proxy server. The default port number is 13111.

- **UDP port**

  If you need Network Agent to connect to Administration Server through a UDP port, enable the **Use UDP port** option and specify a **UDP port number**. By default, this option is enabled. The default UDP port to connect to Administration Server is 15000.

- **HTTPS port**

  If you need the managed devices to connect to the KSN proxy server through an HTTPS port, enable the **Use HTTPS** option, and then specify a port number in the **HTTPS port** field. By default, this option is disabled. The default HTTPS port to connect to the KSN proxy server is 17111.

Updates (distribution points)

In the **Updates (distribution points)** section, you can enable the downloading diff files feature, so distribution points take updates in the form of diff files from Kaspersky update servers.

Local account management (Linux only)

The **Local account management (Linux only)** section includes three subsections:

- **User certificates management**
- **Add or edit applicable local administrator groups**
- **Upload a reference file to protect the sudoers file on the user's device from changes**

In the **User certificates management** subsection, you can specify which root certificates to install. These certificates can be used, for example, to verify the authenticity of websites or web servers.

- **Install root certificates**

  If this option is enabled, certificates added to the table will be installed on the specified devices.

  If this option is disabled, no certificates will be installed on the specified devices.

  By default, this option is disabled.

- **Add**

  Clicking this button opens a window, in which where you can add a certificate.

  The certificate must be less than 10 MB.

  Kaspersky Security Center supports certificates with CER, CRT, CERT, PEM, and KEY extensions.

In the **Add or edit applicable local administrator groups** subsection, you can manage local administrator groups. These groups are used, for example, when revoking local administrator rights. You can also check the list of privileged user accounts using the **Report on privileged device users (Linux only)**.

- **Add**

  Clicking this button opens a window, where you can add a local administrator group.

- **Edit**

  Clicking this button opens a window, where you can edit the local administrator group.

  This button is available if the check box next to the local administrator group is selected.

- **Delete**

  Clicking this button deletes the selected local administrator group from the table.

  This button is available if the check box next to the local administrator group is selected.

In the **Upload a reference file to protect the sudoers file on the user's device from changes** subsection, you can configure control of the sudoers file. Privileged groups and device users are defined by the sudoers file on the device. The sudoers file is located at `/etc/sudoers`. You can upload a reference sudoers file to protect the sudoers file from changes. This will prevent unwanted changes to the sudoers file.

An invalid reference sudoers file may cause the user's device to malfunction.

- **Control sudoers file**

  If this option is enabled, the sudoers file will be replaced by the current reference sudoers file.

  If this option is disabled, the sudoers file will remain unchanged.

  By default, this option is disabled.

- **Reference sudoers file**

  This field displays the name of the uploaded reference sudoers file.

- **Upload**

  Clicking this button opens a window, where you can upload a reference sudoers file.

- **Current reference sudoers file**

  Clicking this button shows the contents of the current sudoers file.

Revision history

On the **Revision history** tab, you can:

- View and save the history of policy revisions.
- Roll back to a policy revision.
- Add and edit policy revision descriptions.

Usage of Network Agent for Windows, Linux, and macOS: Comparison

The Network Agent usage varies depending on the operating system of the device. The Network Agent policy and installation package settings also differ depending on the operating system. The table below compares Network Agent features and usage scenarios available for Windows, Linux, and macOS operating systems.

Network Agent feature comparison

Network Agent feature	Windows	Linux	macOS
Installation			
Installing by cloning an image of the administrator's hard drive with the operating system and Network Agent using third-party tools	✓	✓	✓
Installing with third-party tools for remote installation of applications	✓	✓	✓
Installing manually, by running application installers on devices	✓	✓	✓
Installing Network Agent in silent mode	✓	✓	✓
Manually connecting a client device to the Administration Server. klmover utility	✓	✓	✓
Automatic installing of updates and patches for Open Single Management Platform components	✓	✓	—
Automatic distributing of a key	✓	✓	✓
Forced synchronization	✓	✓	✓ Except a push server
Distribution point			
Using as distribution point	✓	✓	✓

Network Agent feature	Windows	Linux	macOS
Calculating the number and configuration of distribution points	✓	✓ Without using Network Location Awareness (NLA).	✓ Without using Network Location Awareness (NLA).
Offline model of update download	✓	✓	✓
Network polling	✓ • IP range polling • Domain controller polling (Microsoft Active Directory)	✓ IP range polling Zeroconf polling Domain controller polling (Microsoft Active Directory, Samba 4 Active Directory, FreeIPA, ALD Pro)	—
Running KSN proxy service on a distribution point side	✓	✓	—
Downloading updates via Kaspersky update servers to the distribution points repositories that distribute updates to managed devices	✓	✓	— (If one or more devices running Linux or macOS are within the scope of the Download updates to the repositories of distribution points task, the task completes with the Failed status, even if it has successfully completed on all Windows devices.)
Push installation of applications	✓	Restricted: it is not possible to perform push installation on Windows devices by using Linux distribution points.	Restricted: it is not possible to perform push installation on Windows devices by using macOS distribution points.
Using as a push server	✓	✓	—
Handling third-party applications			
Remote installing of applications on devices	✓	✓	✓ Refer to the following article for details: Remote installing of applications on macOS devices.
Configuring operating system updates in a Network Agent policy	✓	—	—
Viewing information about software vulnerabilities	✓	—	—
Scanning applications for vulnerabilities	✓	—	—
Software updates	✓	—	—
Inventory of software installed on devices	✓	✓	—
Virtual machines			
Installing Network Agent on a virtual machine	✓	✓	✓
Optimization settings for virtual desktop infrastructure (VDI)	✓	✓	✓
Support of dynamic virtual machines	✓	✓	✓
Other			
Auditing actions on a remote client device by using Windows Desktop Sharing	✓	—	—
Monitoring the anti-virus protection status	✓	✓	✓
Managing device restarts	✓	—	—

Network Agent feature	Windows	Linux	macOS
Support of file system rollback	✓	✓	✓
Using a Network Agent as connection gateway	✓	✓	✓
Connection Manager	✓	✓	✓
Connection profiles for out-of-office users	✓	✓	✓
Network Agent switching from one Administration Server to another (automatically by network location)	✓	✓	✓
Checking the connection between a client device and the Administration Server. klnagchk utility	✓	✓	✓
Remotely connecting to the desktop of a client device	✓	✓	✓ By using the Virtual Network Computing (VNC) system.
Downloading a stand-alone installation package through the Migration wizard	✓	✓	✓
Viewing information about the hardware of client devices	✓	✓ Information about the hardware of Linux devices sent to the Administration Server by Network Agent installed on these devices is limited to the information specified in the description of the managed device settings.	✓ Information about the hardware of macOS devices sent to the Administration Server by Network Agent installed on these devices is limited to the information specified in the description of the managed device settings.

Comparison of Network Agent settings by operating systems

The table below shows which Network Agent policy settings are available depending on the operating system of the managed device where Network Agent was installed.

Network Agent settings: comparison by operating systems

Settings section	Windows	Linux	macOS
General	✓	✓	✓
Event configuration	✓	✓	✓
Settings	✓	✓ The following options are available: • **Distribute files through distribution points only** • **Maximum size of event queue, in MB** • **Application is allowed to retrieve policy's extended data on device**	✓
Repositories	✓	✓ The following options are available: • **Details of installed applications** • **Hardware registry details**	✓ The **Hardware registry details** option is available.

Settings section	Windows	Linux	macOS
Connectivity → Network	✓	✓ Except the **Open Network Agent ports in Microsoft Windows Firewall** option.	✓
Connectivity → Connection profiles	✓	✓	✓
Connectivity → Connection schedule	✓	✓	✓
Network polling by distribution points	✓ The following options are available: • **Windows network** • **IP ranges** • **Domain controllers**	✓ The following options are available: • **Zeroconf** • **IP ranges** • **Domain controllers**	—
Network settings for distribution points	✓	✓	✓
KSN Proxy (distribution points)	✓	✓	—
Updates (distribution points)	✓	✓	—
Revision history	✓	✓	✓

Manual setup of the Kaspersky Endpoint Security policy

This section provides recommendations on how to configure the Kaspersky Endpoint Security policy. You can perform setup in the policy properties window. When you edit a setting, click the lock icon to the right of the relevant group of settings to apply the specified values to a workstation.

Configuring Kaspersky Security Network

Kaspersky Security Network (KSN) is the infrastructure of cloud services that contains information about the reputation of files, web resources, and software. Kaspersky Security Network enables Kaspersky Endpoint Security for Windows to respond faster to different kinds of threats, enhances the performance of the protection components, and decreases the likelihood of false positives. For more information about Kaspersky Security Network, see the Kaspersky Endpoint Security for Windows Help.

To specify recommended KSN settings:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy of Kaspersky Endpoint Security for Windows.

   The properties window of the selected policy opens.
3. In the policy properties, go to **Application settings** → **Advanced Threat Protection** → **Kaspersky Security Network**.

4. Make sure that the **Kaspersky Security Network** option is enabled. Using this option helps to redistribute and optimize traffic on the network.

   If you use Managed Detection and Response, you must enable **Kaspersky Security Network** option for the distribution point and enable extended KSN mode.

5. Enable use of KSN servers if the KSN proxy service is not available. KSN servers may be located either on the side of Kaspersky (when KSN is used) or on the side of third parties (when KPSN is used).

6. Click **OK**.

The recommended KSN settings are specified.

Checking the list of the networks protected by Firewall

Make sure that Kaspersky Endpoint Security for Windows Firewall protects all your networks. By default, Firewall protects networks with the following types of connection:

- **Public network**. Security applications, firewalls, or filters do not protect devices in such a network.
- **Local network**. Access to files and printers is restricted for devices in this network.
- **Trusted network**. Devices in such a network are protected from attacks and unauthorized access to files and data.

If you configured a custom network, make sure that Firewall protects it. For this purpose, check the list of the networks in the Kaspersky Endpoint Security for Windows policy properties. The list may not contain all the networks.

For more information about Firewall, see the Kaspersky Endpoint Security for Windows Help.

To check the list of networks:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy of Kaspersky Endpoint Security for Windows.

   The properties window of the selected policy opens.
3. In the policy properties, go to **Application settings** → **Essential Threat Protection** → **Firewall**.
4. Under **Available networks**, click the **Network settings** link.

   The **Network connections** window opens. This window displays the list of networks.
5. If the list has a missing network, add it.

Disabling the scan of network drives

When Kaspersky Endpoint Security for Windows scans network drives, this can place a significant load on them. It is more convenient to perform indirect scanning on file servers.

You can disable scanning of network drives in the Kaspersky Endpoint Security for Windows policy properties. For a description of these policy properties, see the Kaspersky Endpoint Security for Windows Help.

To disable scanning of network drives:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy of Kaspersky Endpoint Security for Windows.

   The properties window of the selected policy opens.
3. In the policy properties, go to **Application settings** → **Essential Threat Protection** → **File Threat Protection**.
4. Under **Protection scope**, disable the **All network drives** option.
5. Click **OK**.

Scanning of network drives is disabled.

Excluding software details from the Administration Server memory

We recommend that Administration Server does not save information about software modules that are started on the network devices. As a result, the Administration Server memory does not overrun.

You can disable saving this information in the Kaspersky Endpoint Security for Windows policy properties.

To disable saving information about installed software modules:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy of Kaspersky Endpoint Security for Windows.

   The properties window of the selected policy opens.
3. In the policy properties, go to **Application settings** → **General Settings** → **Reports and Storage**.
4. Under **Data transfer to Administration Server**, disable the **About started applications** check box if it is still enabled in the top-level policy.

   When this check box is selected, the Administration Server database saves information about all versions of all software modules on the networked devices. This information may require a significant amount of disk space in the Open Single Management Platform database (dozens of gigabytes).

The information about installed software modules is no longer saved to the Administration Server database.

Configuring access to the Kaspersky Endpoint Security for Windows interface on workstations

If the threat protection on the organization's network must be managed in centralized mode through Open Single Management Platform, specify the interface settings in the Kaspersky Endpoint Security for Windows policy properties, as described below. As a result, you will prevent unauthorized access to Kaspersky Endpoint Security for Windows on workstations and the changing of Kaspersky Endpoint Security for Windows settings.

For a description of these policy properties, see the Kaspersky Endpoint Security for Windows Help.

To specify recommended interface settings:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy of Kaspersky Endpoint Security for Windows.

   The properties window of the selected policy opens.
3. In the policy properties, go to **Application settings** → **General settings** → **Interface**.
4. Under **Interaction with user**, select the **Do not display user interface** option. This disables the display of the Kaspersky Endpoint Security for Windows user interface on workstations, so their users cannot change the settings of Kaspersky Endpoint Security for Windows.
5. Under **Password protection**, enable the toggle switch. This reduces the risk of unauthorized or unintended changes in the settings of Kaspersky Endpoint Security for Windows on workstations.

The recommended settings for the interface of Kaspersky Endpoint Security for Windows are specified.

Configuring the registration of policy events in the Administration Server database

To avoid Administration Server database overflow, we recommend that you save only important events to the database. For events that you consider unimportant, you can reduce or disable the storage period.

To configure the event storage settings:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the name of the required policy.

   The properties window of the selected policy opens.
3. Go to the **Event configuration** tab, and then click the name of the event type for which you want to configure the registration in the database.
4. In the right pane that opens, do one of the following:
   - If you want to change the storage period for the event type, make sure that the **Store in the Administration Server database for (days)** toggle button is turned on, and then enter the required number of days for the event type to be stored.
   - If you do not want to store the event type in the in the Administration Server database, turn off the **Store in the Administration Server database for (days)** toggle button.
5. Click **OK**, and then after the right pane is closed, click the **Save** button.

The policy properties window is closed, and the setting that you configured is applied.

Manual setup of the group update task for Kaspersky Endpoint Security

The optimal and recommended schedule option for Kaspersky Endpoint Security is **When new updates are downloaded to the repository** when the **Use automatically randomized delay for task starts** check box is selected.

Kaspersky Security Network (KSN)

This section describes how to use an online service infrastructure named Kaspersky Security Network (KSN). The section provides the details on KSN, as well as instructions on how to enable KSN, configure access to KSN, and view the statistics of the use of KSN proxy server.

> Updates functionality (including providing anti-virus signature updates and codebase updates), as well as KSN functionality may not be available in the software in the U.S.

About KSN

Kaspersky Security Network (KSN) is an online service infrastructure that provides access to the online Knowledge Base of Kaspersky, which contains information about the reputation of files, web resources, and software. The use of data from Kaspersky Security Network ensures faster responses by Kaspersky applications to threats, improves the effectiveness of some protection components, and reduces the risk of false positives. KSN allows you to use Kaspersky reputation databases to retrieve information about applications installed on managed devices.

By participating in KSN, you agree to send to Kaspersky in automatic mode information about the operation of Kaspersky applications installed on client devices that are managed through Open Single Management Platform. Information is transferred in accordance with the current KSN access settings.

Open Single Management Platform supports the following KSN infrastructure solutions:

- *Global KSN* is a solution that allows you to exchange information with Kaspersky Security Network. If you participate in KSN, you agree to send to Kaspersky, in automatic mode, information about the operation of Kaspersky applications installed on client devices that are managed through Open Single Management Platform. Information is transferred in accordance with the current KSN access settings. Kaspersky analysts additionally analyze received information and include it in the reputation and statistical databases of Kaspersky Security Network. Open Single Management Platform uses this solution by default.

- *Kaspersky Private Security Network (KPSN)* is a solution that allows users of devices with Kaspersky applications installed to obtain access to reputation databases of Kaspersky Security Network, and other statistical data, without sending data to KSN from their own computers. KPSN is designed for corporate customers who are unable to participate in Kaspersky Security Network for any of the following reasons:

  - User devices are not connected to the internet.

  - Transmission of any data outside the country or outside the corporate LAN is prohibited by law or restricted by corporate security policies.

You can set up access settings of Kaspersky Private Security Network in the **KSN Proxy** section of the Administration Server properties window.

You can start or stop using KSN at any moment.

You use KSN in accordance with the KSN Statement that you read and accept when you enable KSN. If the KSN Statement is updated, it is displayed to you when you update or upgrade Administration Server. You can accept the updated KSN Statement or decline it. If you decline it, you keep using KSN in accordance with the previous version of KSN Statement that you accepted before.

When KSN is enabled, Open Single Management Platform checks if the KSN servers are accessible. If access to the servers using system DNS is not possible, the application uses public DNS servers. This is necessary to make sure the level of security is maintained for the managed devices.

Client devices managed by the Administration Server interact with KSN through KSN proxy server. KSN proxy server provides the following features:

- Client devices can send requests to KSN and transfer information to KSN even if they do not have direct access to the internet.
- The KSN proxy server caches processed data, thus reducing the load on the outbound channel and the time period spent for waiting for information requested by a client device.

You can configure the KSN proxy server in the **KSN Proxy** section of the Administration Server properties window.

Setting up access to KSN

You can set up access to Kaspersky Security Network (KSN) on the Administration Server and on a distribution point.

KSN functionality is only available with current Kaspersky EDR Expert license.

To set up Administration Server access to KSN:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **KSN Proxy** section.

3. Switch the toggle button to the **Enable KSN Proxy on Administration Server Enabled** position.

   Data is sent from client devices to KSN in accordance with the Kaspersky Endpoint Security policy, which is active on those client devices. If this check box is cleared, no data will be sent to KSN from the Administration Server and client devices through Open Single Management Platform. However, client devices can send data to KSN directly (bypassing Open Single Management Platform), in accordance with their respective settings. The Kaspersky Endpoint Security policy, which is active on client devices, determines which data will be sent directly (bypassing Open Single Management Platform) from those devices to KSN.

4. Switch the toggle button to the **Use Kaspersky Security Network Enabled** position.

   If this option is enabled, client devices send patch installation results to Kaspersky. When enabling this option, make sure to read and accept the terms of the KSN Statement.

   If you are using KPSN, switch the toggle button to the **Use Kaspersky Private Security Network Enabled** position and click the **Select file with KSN Proxy settings** button to download the settings of KPSN (files with the extensions pkcs7 and pem). After the settings are downloaded, the interface displays the provider's name and contacts, as well as the creation date of the file with the settings of KPSN.

When you switch the toggle button to the **Use Kaspersky Private Security Network Enabled** position, a message appears with details about KPSN.

The following Kaspersky applications support KPSN:

- Open Single Management Platform
- Kaspersky Endpoint Security for Linux
- Kaspersky Endpoint Security for Windows

If you enable KPSN in Open Single Management Platform, these applications receive information about supporting KPSN. In the settings window of the application, in the **Kaspersky Security Network** subsection of the **Advanced Threat Protection** section, the information about selected KSN provider is displayed — KSN or KPSN.

> Open Single Management Platform does not send any statistical data to Kaspersky Security Network if KPSN is configured in the **KSN Proxy** section of the Administration Server properties window.

5. If you have the proxy server settings configured in the Administration Server properties, but your network architecture requires that you use KPSN directly, enable the **Ignore proxy server settings when connecting to KPSN** option. Otherwise, requests from the managed applications cannot reach KPSN.

6. Under **Connection settings**, configure the Administration Server connection to the KSN proxy service:

   - The TCP port 13111 is used for connecting to the KSN proxy server. For the root Administration Server, this port number cannot be changed.
   - If you want the Administration Server to connect to the KSN proxy server through a UDP port, enable the **Use UDP port** option. By default, this option is disabled, and TCP port is used. If this option is enabled, the UDP port 15111 is used by default. For the root Administration Server, this port number cannot be changed.

7. Switch the toggle button to the **Connect secondary Administration Servers to KSN through the primary Administration Server Enabled** position.

   If this option is enabled, secondary Administration Servers use the primary Administration Server as the KSN proxy server. If this option is disabled, secondary Administration Servers connect to KSN on their own. In this case, managed devices use secondary Administration Servers as KSN proxy servers.

   > Secondary Administration Servers use the primary Administration Server as a proxy server if in the right pane of the **KSN Proxy** section, in the properties of secondary Administration Servers the toggle button is switched to the **Enable KSN Proxy on Administration Server Enabled** position.

8. Click the **Save** button.

The KSN access settings will be saved.

You can also set up distribution point access to KSN, for example, if you want to reduce the load on the Administration Server. The distribution point that acts as a KSN proxy server sends KSN requests from managed devices to Kaspersky directly, without using the Administration Server.

To set up distribution point access to Kaspersky Security Network (KSN):

1. Make sure that the distribution point is assigned manually.
2. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.
3. On the **General** tab, select the **Distribution points** section.
4. Click the name of the distribution point to open its properties window.
5. In the distribution point properties window, in the **KSN Proxy** section, enable the **Enable KSN Proxy on the distribution point side** option, and then enable the **Access KSN Cloud/KPSN directly over the internet** option.
6. Click **OK**.

The distribution point will act as a KSN proxy server.

Please note that the distribution point does not support managed device authentication by using the NTLM protocol.

Enabling and disabling the usage of KSN

To enable the usage of KSN:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.
2. On the **General** tab, select the **KSN Proxy** section.
3. Switch the toggle button to the **Enable KSN Proxy on Administration Server Enabled** position.

The KSN proxy server is enabled and sends data to KSN to increase the efficiency of Kaspersky Security Center components and improve the performance of Kaspersky applications.

1. Depending on the KSN infrastructure solution that you are using, enable the corresponding toggle buttons.

   - If you are using Global KSN, switch the toggle button to the **Use Kaspersky Security Network Enabled** position.

     Sending data to KSN is now available. When enabling this option, you have to read and accept the terms of the KSN Statement.

   - If you are using KPSN, switch the toggle button to the **Use Kaspersky Private Security Network Enabled** position, and then click the **Select file with KSN Proxy settings** button to download the settings of KPSN (files with the extensions pkcs7 and pem). After the settings are downloaded, the interface displays the provider's name and contacts, as well as the creation date of the file with the settings of KPSN.

     When you switch the toggle button to the **Use Kaspersky Private Security Network Enabled** position, a message appears with details about KPSN.
2. Click the **Save** button.

To disable the usage of KSN:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **KSN Proxy** section.

3. Switch the toggle button to the **Enable KSN Proxy on Administration Server Disabled** position to disable the KSN proxy service.

4. Click the **Save** button.

Configuring the connection to a local reputation database of KPSN

Kaspersky Next XDR Expert can save information about Sandbox component alerts to the local reputation database of KPSN. In this case, the *Untrusted* status is assigned to objects. Data of local reputation databases is available only to corporate LAN computers.

The application allows users with the following roles to view and edit Kaspersky Private Security Network API integration settings:

- Main administrator
- SOC administrator
- Tenant administrator (settings editing is available only for a root tenant, for other tenants in the hierarchy, settings are applied automatically)

Users with the Auditor role can only view Kaspersky Private Security Network API integration settings.

To configure the connection of Kaspersky Next XDR Expert to a local KPSN reputation database:

1. In the main menu, go to **Settings** → **Tenants**.

2. Click the **Root tenant** and go to the **Settings** → **KPSN reputation database** subsection.

3. In the **Asset** field, specify the IP address of the KPSN server on which the local reputation database of KPSN is stored.

4. Click **Upload** to the right of the **Public key for user certificate** field and select the required .pem certificate.

5. Click **Upload** to the right of the **Private key for user certificate** field and select the required .pem certificate.

6. Click **Upload** to the right of the **Trusted root certificate for KPSN server** field and select the required .pem certificate.

7. Click **Save**.

   The connection to the local reputation database of KPSN will be configured.

Viewing the accepted KSN Statement

When you enable Kaspersky Security Network (KSN), you must read and accept the KSN Statement. You can view the accepted KSN Statement at any time.

To view the accepted KSN Statement:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **KSN Proxy** section.

3. Click the **View Kaspersky Security Network Statement** link.

In the window that opens, you can view the text of the accepted KSN Statement.

Accepting an updated KSN Statement

You use KSN in accordance with the KSN Statement that you read and accept when you enable KSN. If the KSN Statement is updated, it is displayed to you when you upgrade a version of Administration Server. You can accept the updated KSN Statement or decline it. If you decline it, you will continue using KSN in accordance with the version of the KSN Statement that you previously accepted.

After upgrading a version of Administration Server, the updated KSN Statement is displayed automatically. If you decline the updated KSN Statement, you can still view and accept it later.

To view and then accept or decline an updated KSN Statement:

1. Click the **View notifications** link in the upper-right corner of the main application window.

   The **Notifications** window opens.

2. Click the **View the updated KSN Statement** link.

   The **Kaspersky Security Network Statement update** window opens.

3. Read the KSN Statement, and then make your decision by clicking one of the following buttons:

   - **I accept the updated KSN Statement**
   - **Use KSN under the old Statement**

Depending on your choice, KSN keeps working in accordance with the terms of the current or updated KSN Statement. You can view the text of the accepted KSN Statement in the properties of Administration Server at any time.

Checking whether the distribution point works as KSN proxy server

On a managed device assigned to work as a distribution point, you can enable Kaspersky Security Network (KSN) Proxy. A managed device works as the KSN proxy server when the ksnproxy service is running on the device. You can check, turn on, or turn off this service on the device locally.

You can assign a Windows-based or a Linux-based device as a distribution point. The method of distribution point checking depends on the operating system of this distribution point.

To check whether the Linux-based distribution point works as KSN proxy server:

1. On the distribution point device, run the `ps aux` command to display the list of running processes.

2. In the list of running processes, check whether the `/opt/kaspersky/klnagent64/sbin/ksnproxy` process is running.

If `/opt/kaspersky/klnagent64/sbin/ksnproxy` process is running, then Network Agent on the device participates in Kaspersky Security Network and works as the KSN proxy server for the managed devices included in the scope of the distribution point.

To check whether the Windows-based distribution point works as KSN proxy server:

1. On the distribution point device, in Windows, open **Services** (**All Programs** → **Administrative Tools** → **Services**).

2. In the list of services, check whether the ksnproxy service is running.

   If the ksnproxy service is running, then Network Agent on the device participates in Kaspersky Security Network and works as KSN proxy server for the managed devices included in the scope of the distribution point.

If you want, you may turn off the ksnproxy service. In this case, Network Agent on the distribution point stops participating in Kaspersky Security Network. This requires local administrator rights.

Viewing the statistics of KSN proxy server

OSMP Console allows you to view usage statistics of KSN proxy server.

Statistics includes the following parameters:

- Connection status to KSN
- Number of cache records
- Number of packages processed in cache
- Number of received packages

If you use KPSN as an infrastructure solution, usage statistics of KPSN will be displayed.

To view the statistics:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **KSN Proxy** section, and then open the **Statistics** tab.

   This tab displays the actual statistics of the use of the KSN proxy server. Statistics refreshes every 15 seconds.

You can copy statistics by clicking the **Copy to clipboard** button.

Managing endpoint protection tasks

This section describes tasks used by Open Single Management Platform.

About tasks

Open Single Management Platform manages Kaspersky security applications installed on devices by creating and running *tasks*. Tasks are required for installing, launching, and stopping applications, scanning files, updating databases and software modules, and performing other actions on applications.

Tasks for a specific application can be created using OSMP Console only if the management plug-in for that application is installed on OSMP Console Server.

Tasks can be performed on the Administration Server and on devices.

The tasks that are performed on the Administration Server include the following:

- Automatic distribution of reports
- Downloading of updates to the repository
- Backup of Administration Server data
- Maintenance of the database

The following types of tasks are performed on devices:

- *Local tasks*—Tasks that are performed on a specific device

  Local tasks can be modified either by the administrator, using OSMP Console, or by the user of a remote device (for example, through the security application interface). If a local task has been modified simultaneously by the administrator and the user of a managed device, the changes made by the administrator will take effect because they have a higher priority.

- *Group tasks*—Tasks that are performed on all devices of a specific group

  Unless otherwise specified in the task properties, a group task also affects all subgroups of the selected group. A group task also affects (optionally) devices that have been connected to secondary and virtual Administration Servers deployed in the group or any of its subgroups.

- *Global tasks*—Tasks that are performed on a set of devices, regardless of whether they are included in any group.

For each application, you can create any number of group tasks, global tasks, or local tasks.

You can make changes to the settings of tasks, view the progress of tasks, and copy, export, import, and delete tasks.

A task is started on a device only if the application for which the task was created is running.

Execution results of tasks are saved in the operating system event log on each device, in the operating system event log on the Administration Server, and in the Administration Server database.

> Do not include private data in task settings. For example, avoid specifying the domain administrator password.

About task scope

The *scope of a task* is the set of devices on which the task is performed. The types of scope are as follows:

- For a *local task*, the scope is the device itself.
- For an *Administration Server task*, the scope is the Administration Server.
- For a *group task*, the scope is the list of devices included in the group.

When creating a *global task*, you can use the following methods to specify its scope:

- Specifying certain devices manually.

  You can use an IP address (or IP range) or DNS name as the device address. You can also select networked devices detected by Administration Server.

- Importing a list of devices from a .txt file with the device addresses to be added (each address must be placed on an individual line).

  If you import a list of devices from a file or create a list manually, and if devices are identified by their names, the list can only contain devices for which information has already been entered into the Administration Server database. Moreover, the information must have been entered when those devices were connected or during device discovery.

- Specifying a device selection.

  Over time, the scope of a task changes as the set of devices included in the selection change. A selection of devices can be made on the basis of device attributes, including software installed on a device, and on the basis of tags assigned to devices. Device selection is the most flexible way to specify the scope of a task.

  Tasks for device selections are always run on a schedule by the Administration Server. These tasks cannot be run on devices that lack connection to the Administration Server. Tasks whose scope is specified by using other methods are run directly on devices and therefore do not depend on the device connection to the Administration Server.

  Tasks for device selections are not run on the local time of a device; instead, they are run on the local time of the Administration Server. Tasks whose scope is specified by using other methods are run on the local time of a device.

Creating a task

To create a task:

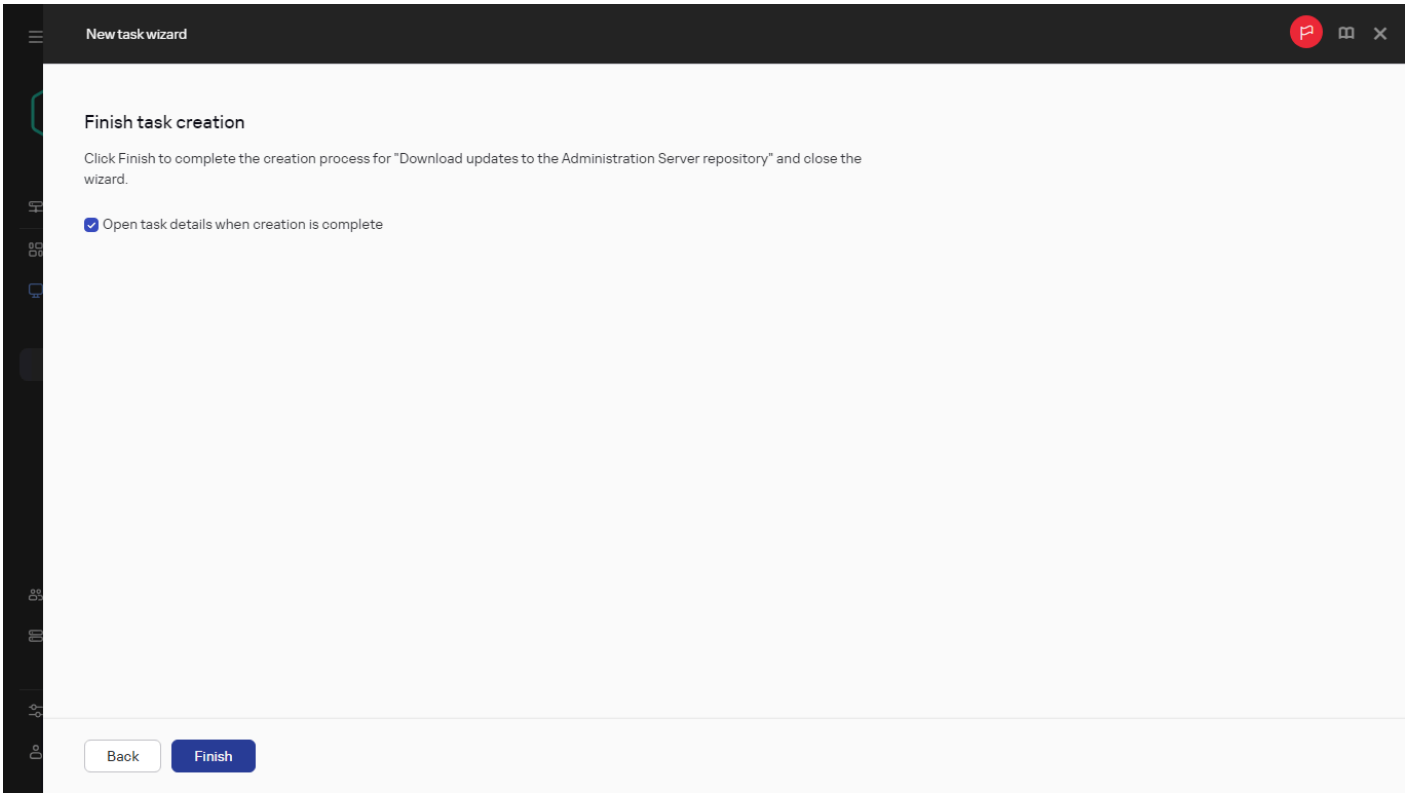
1. In the main menu, go to **Asset management** → **Tasks**.
2. Click **Add**.

   The New task wizard starts. Follow its instructions.

3. If you want to modify the default task settings, enable the **Open task details when creation is complete** option on the **Finish task creation** page. If you do not enable this option, the task is created with the default settings. You can modify the default settings later, at any time.

4. Click the **Finish** button.

The task is created and displayed in the list of tasks.



Finishing task creation

To create a new task assigned to the selected devices:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The list of managed devices is displayed.

2. In the list of managed devices, select check boxes next to the devices to run the task for them. You can use the search and filter functions to find the devices you're looking for.

3. Click the **Run task** button, and then select **Add a new task**.

   The New task wizard starts.

   On the first step of the wizard, you can remove the devices selected to include in the task scope. Follow the wizard instructions.

4. Click the **Finish** button.

The task is created for the selected devices.

Starting a task manually

The application starts tasks according to the schedule settings specified in the properties of each task. You can start a task manually at any time from the task list. Alternatively, you can select devices in the **Assets (Devices)** list, and then start an existing task for them.

To start a task manually:

1. In the main menu, go to **Asset management** → **Tasks**.
2. In the task list, select the check box next to the task that you want to start.
3. Click the **Start** button.

   The task starts. You can check the task status in the **Status** column or by clicking the **Result** button.

Starting a task for selected devices

You can select one or more client devices in the list of devices, and then launch a previously created task for them. This allows you to run tasks created earlier for a specific set of devices.

> This changes the devices to which the task was assigned to the list of devices that you select when you run the task.

To start a task for selected devices:

1. In the main menu, go to **Asset management** → **Assets (Devices)**. The list of managed devices is displayed.
2. In the list of managed devices, use the check boxes to select the devices to run the task for them. You can use the search and filter functions to find the devices you're looking for.
3. Click the **Run task** button, and then select **Apply existing task**.

   The list of the existing tasks is displayed.
4. The selected devices are displayed above the task list. If necessary, you can remove a device from this list. You can delete all but one device.
5. Select the desired task in the list. You can use the search box above the list to search for the desired task by name. Only one task can be selected.
6. Click **Save and start task**.

   The selected task is immediately started for the selected devices. The scheduled start settings in the task are not changed.

Viewing the task list

You can view the list of tasks that are created in Open Single Management Platform.

To view the list of tasks,

In the main menu, go to **Asset management** → **Tasks**.

The list of tasks is displayed. The tasks are grouped by the names of applications to which they are related. For example, the *Install application remotely* task is related to the Administration Server, and the *Update* task refers to Kaspersky Endpoint Security.

To view properties of a task,

Click the name of the task.

The task properties window is displayed with several named tabs. For example, the **Task type** is displayed on the **General** tab, and the task schedule—on the **Schedule** tab.

General task settings

This section contains the settings that you can view and configure for most of your tasks. The list of settings available depends on the task you are configuring.

Settings specified during task creation

You can specify the following settings when creating a task. Some of these settings can also be modified in the properties of the created task.

- Operating system restart settings:
  - **Do not restart the device**

    Client devices are not restarted automatically after the operation. To complete the operation, you must restart a device (for example, manually or through a device management task). Information about the required restart is saved in the task results and in the device status. This option is suitable for tasks on servers and other devices where continuous operation is critical.

  - **Restart the device**

    Client devices are always restarted automatically if a restart is required for completion of the operation. This option is useful for tasks on devices that provide for regular pauses in their operation (shutdown or restart).

  - **Force closure of applications in blocked sessions**

    Running applications may prevent a restart of the client device. For example, if a document is being edited in a word processing application and is not saved, the application does not allow the device to restart.

    If this option is enabled, such applications on a locked device are forced to close before the device restart. As a result, users may lose their unsaved changes.

If this option is disabled, a locked device is not restarted. The task status on this device states that a device restart is required. Users have to manually close all applications running on locked devices and restart these devices.

By default, this option is disabled.

- Task scheduling settings:

The types of schedule may vary depending on the task.

Some types of schedule may be unavailable for other Kaspersky applications.

- **Start task setting:**
  - **Every N hours**

    The task runs regularly, with the specified interval in hours, starting from the specified date and time.

    By default, the task runs every 6 hours, starting from the current system date and time.

  - **Every N days**

    The task runs regularly, with the specified interval in days. Additionally, you can specify a date and time of the first task run. These additional options become available, if they are supported by the application for which you create the task.

    By default, the task runs every day, starting from the current system date and time.

  - **Every N minutes**

    The task runs regularly, with the specified interval in minutes, starting from the specified time on the day that the task is created.

    By default, the task runs every 30 minutes, starting from the current system time.

  - **By days of week**

    The task runs regularly, on the specified days of the week, at the specified time.

    By default, the task runs every Friday at 6:00:00 PM.

  - **Monthly**

    The task runs regularly, on the specified day of the month, at the specified time.

    In months that lack the specified day, the task runs on the last day.

    By default, the task runs on the first day of each month, at the current system time.

  - **Manually**

    The task does not run automatically. You can only start it manually.

By default, this option is selected.

- **Once**

  The task runs once, on the specified date and time (by default, on the day when the task was created).

- **Immediately**

  The task runs immediately after its settings are saved.

- **Every month on specified days of selected weeks**

  The task runs regularly, on the specified days of each month, at the specified time.

  By default, the start time is 18:00, and no days of the month are selected.

  Note that you select neither a specific date in the month, nor the number of the week (first, second week of the month) but the sequence number of the weekday within a month. For example, if you set the cursor in the **Tu** cell of the **First** row, it means the task will run every first Tuesday of each new month.

  You can select several days of week.

- **When new updates are downloaded to the repository**

  The task runs after updates are downloaded to the repository. For example, you may want to use this schedule for the *Update* task.

- **On completing another task**

  The current task starts after another task completes. This option only works if both tasks are assigned to the same devices. For example, you may want to run the *Manage devices* task with the **Turn on the device** option and, after it completes, run the *Virus scan* task as a triggering task.

  You have to select the triggering task from the table and the status with which this task must complete (**Completed successfully** or **Failed**).

  If necessary, you can search, sort, and filter the tasks in the table as follows:

  - Enter the task name in the search field, to search the task by its name.
  - Click the sort icon to sort the tasks by name.

    By default, the tasks are sorted in alphabetical ascending order.
  - Click the filter icon, and in the window that opens, filter the tasks by group, and then click the **Apply** button.

- **Run missed tasks**

  This option determines the behavior of a task if a client device is not visible on the network when the task is about to start.

  If this option is enabled, the system attempts to start the task the next time the Kaspersky application is run on the client device. If the task schedule is **Manually**, **Once** or **Immediately**, the task is started immediately after the device becomes visible on the network or immediately after the device is included in the task scope.

  If this option is disabled, only scheduled tasks run on client devices. For **Manually**, **Once** and **Immediately** schedule, tasks run only on those client devices that are visible on the network. For example, you may want to disable this option for a resource-consuming task that you want to run only outside of business hours.

  By default, this option is disabled.

- **Use automatically randomized delay for task starts**

  If this option is enabled, the task is started on client devices randomly within a specified time interval, that is, *distributed task start*. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  The distributed start time is calculated automatically when a task is created, depending on the number of client devices to which the task is assigned. Later, the task is always started on the calculated start time. However, when task settings are edited or the task is started manually, the calculated value of the task start time changes.

  If this option is disabled, the task starts on client devices according to the schedule.

- **Use automatically randomized delay for task starts within an interval of**

  If this option is enabled, the task is started on client devices randomly within the specified time interval. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  If this option is disabled, the task starts on client devices according to the schedule.

  By default, this option is disabled. The default time interval is one minute.

- Devices to which the task will be assigned:

  - **Select networked devices detected by Administration Server**

    The task is assigned to specific devices. The specific devices can include devices in administration groups as well as unassigned devices.

    For example, you may want to use this option in a task of installing Network Agent on unassigned devices.

  - **Specify device addresses manually or import addresses from list**

    You can specify DNS names, IP addresses, and IP subnets of devices to which you want to assign the task.

    You may want to use this option to execute a task for a specific subnet. For example, you may want to install a certain application on devices of accountants or to scan devices in a subnet that is probably infected.

- **Assign task to a device selection**

  The task is assigned to devices included in a device selection. You can specify one of the existing selections.

  For example, you may want to use this option to run a task on devices with a specific operating system version.

- **Assign task to an administration group**

  The task is assigned to devices included in an administration group. You can specify one of the existing groups or create a new one.

  For example, you may want to use this option to run a task of sending a message to users if the message is specific for devices included in a specific administration group.

  If a task is assigned to an administration group, the **Security** tab is not displayed in the task properties window because group tasks are subject to the security settings of the groups to which they apply.

- Account settings:

  - **Default account**

    The task will be run under the same account as the application that performs this task.

    By default, this option is selected.

  - **Specify an account**

    Fill in the **Account** and **Password** fields to specify the details of an account under which the task is run. The account must have sufficient rights for this task.

  - **Account**

    Account under which the task is run.

  - **Password**

    Password of the account under which the task will be run.

Settings specified after task creation

You can specify the following settings only after a task is created.

- Group task settings:

  - **Distribute to subgroups**

    This option is only available in the settings of the group tasks.

When this option is enabled, the task scope includes:

- The administration group that you selected while creating the task.
- The administration groups subordinate to the selected administration group at any level down by the group hierarchy.

When this option is disabled, the task scope includes only the administration group that you selected while creating the task.

By default, this option is enabled.

- **Distribute to secondary and virtual Administration Servers**

When this option is enabled, the task that is effective on the primary Administration Server is also applied on the secondary Administration Servers (including virtual ones). If a task of the same type already exists on the secondary Administration Server, both tasks are applied on the secondary Administration Server—the existing one and the one that is inherited from the primary Administration Server.

This option is only available when the **Distribute to subgroups** option is enabled.

By default, this option is disabled.

- Advanced scheduling settings:

- **Turn on devices by using the Wake-on-LAN function before starting the task**

The operating system on the device starts at the specified time before the task is started. The default time period is five minutes.

Enable this option if you want the task to run on all of the client devices from the task scope, including those devices that are turned off when the task is about to start.

If you want the device to be automatically turned off after the task is completed, enable the **Shut down the devices after completing the task** option. This option can be found in the same window.

By default, this option is disabled.

- **Shut down the devices after completing the task**

For example, you may want to enable this option for an install update task that installs updates to client devices each Friday after business hours, and then turns off these devices for the weekend.

By default, this option is disabled.

- **Stop the task if it runs longer than**

After the specified time period expires, the task is stopped automatically, whether it is completed or not.

Enable this option if you want to interrupt (or stop) tasks that take too long to execute.

By default, this option is disabled. The default task execution time is 120 minutes.

- Notification settings:
  - **Store task history** block:
    - **Store in the Administration Server database for (days)**

      Application events related to execution of the task on all client devices from the task scope are stored on the Administration Server during the specified number of days. When this period elapses, the information is deleted from the Administration Server.

      By default, this option is enabled.

    - **Store in the OS event log on device**

      Application events related to execution of the task are stored locally in the Syslog Event Log of each client device.

      By default, this option is disabled.

    - **Store in the OS event log on Administration Server**

      Application events related to execution of the task on all client devices from the task scope are stored centrally in the Syslog Event Log of the Administration Server operating system (OS).

      By default, this option is disabled.

    - **Save all events**

      If this option is selected, all events related to the task are saved to the event logs.

    - **Save events related to task progress**

      If this option is selected, only events related to the task execution are saved to the event logs.

    - **Save only task execution results**

      If this option is selected, only events related to the task results are saved to the event logs.

  - **Notify administrator of task execution results**

    You can select the methods by which administrators receive notifications about task execution results: by email, by SMS, and by running an executable file. To configure notification, click the **Settings** link.

    By default, all notification methods are disabled.

  - **Notify of errors only**

    If this option is enabled, administrators are only notified when a task execution completes with an error.

    If this option is disabled, administrators are notified after every task execution completion.

By default, this option is enabled.

- Security settings.
- Task scope settings.

  Depending on how the task scope is determined, the following settings are present:

  - **Devices**

    If the scope of a task is determined by an administration group, you can view this group. No changes are available here. However, you can set **Exclusions from task scope**.

    If the scope of a task is determined by a list of devices, you can modify this list by adding and removing devices.

  - **Device selection**

    You can change the device selection to which the task is applied.

  - **Exclusions from task scope**

    You can specify groups of devices to which the task is not applied. Groups to be excluded can only be subgroups of the administration group to which the task is applied.

- **Revision history**.

Exporting a task

Open Single Management Platform allows you to save a task and its settings to a KLT file. You can use this KLT file to import the saved task both to Kaspersky Security Center Windows and Kaspersky Security Center Linux.

To export a task:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Select the check box next to the task that you want to export.

   You cannot export multiple tasks at the same time. If you select more than one task, the **Export** button will be disabled. Administration Server tasks are also unavailable for export.
3. Click the **Export** button.
4. In the opened **Save as** window, specify the task file name and path. Click the **Save** button.

   The **Save as** window is displayed only if you use Google Chrome, Microsoft Edge, or Opera. If you use another browser, the task file is automatically saved in the **Downloads** folder.

Importing a task

Open Single Management Platform allows you to import a task from a KLT file. The KLT file contains the exported task and its settings.

For correct import of the Kaspersky application task, you have to install the appropriate management plug-in.

To import a task:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click the **Import** button.
3. Click the **Browse** button to choose a task file that you want to import.
4. In the opened window, specify the path to the KLT task file, and then click the **Open** button. Note that you can select only one task file.

   The task processing starts.
5. After the task is processed successfully, select the devices to which you want to assign the task. To do this, select one of the following options:

   - **Assign task to an administration group**

     The task is assigned to devices included in an administration group. You can specify one of the existing groups or create a new one.

     For example, you may want to use this option to run a task of sending a message to users if the message is specific for devices included in a specific administration group.

     If a task is assigned to an administration group, the **Security** tab is not displayed in the task properties window because group tasks are subject to the security settings of the groups to which they apply.

   - **Specify device addresses manually or import addresses from a list**

     You can specify DNS names, IP addresses, and IP subnets of devices to which you want to assign the task.

     You may want to use this option to execute a task for a specific subnet. For example, you may want to install a certain application on devices of accountants or to scan devices in a subnet that is probably infected.

   - **Assign task to a device selection**

     The task is assigned to devices included in a device selection. You can specify one of the existing selections.

     For example, you may want to use this option to run a task on devices with a specific operating system version.
6. Specify the task scope.
7. Click the **Complete** button to finish the task import.

The notification with the import results appears. If the task is imported successfully, you can click the **Details** link to view the task properties.

After a successful import, the task is displayed in the task list. The task settings and schedule are also imported. The task will be started according to its schedule.

> If the newly imported task has an identical name to an existing task, the name of the imported task is expanded with the **(<next sequence number>)** index, for example: **(1)**, **(2)**.

Starting the Change tasks password wizard

For a non-local task, you can specify an account under which the task must be run. You can specify the account during task creation or in the properties of an existing task. If the specified account is used in accordance with security instructions of the organization, these instructions might require changing the account password from time to time. When the account password expires and you set a new one, the tasks will not start until you specify the new valid password in the task properties.

The Change tasks password wizard enables you to automatically replace the old password with the new one in all tasks in which the account is specified. Alternatively, you can change this password manually in the properties of each task.

To start the Change tasks password wizard:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click **Manage credentials of accounts for starting tasks**.

   Follow the instructions of the wizard.

Step 1. Specifying credentials

Specify new credentials that are currently valid in your system. When you switch to the next step of the wizard, Open Single Management Platform checks if the specified account name matches the account name in the properties of each non-local task. If the account names match, the password in the task properties will be automatically replaced with the new one.

To specify the new account, select an option:

- **Use current account**

  > The wizard uses the name of the account under which you are currently signed in to OSMP Console. Then manually specify the account password in the **Current password to use in tasks** field.

- **Specify a different account**

  > Specify the name of the account under which the tasks must be started. Then specify the account password in the **Current password to use in tasks** field.

If you fill in the **Previous password (optional; if you want to replace it with the current one)** field, Open Single Management Platform replaces the password only for those tasks in which both the account name and the old password are found. The replacement is performed automatically. In all other cases you have to choose an action to take in the next step of the wizard.

Step 2. Selecting an action to take

If you did not specify the previous password in the first step of the wizard or if the specified old password has not matched the passwords in the task properties, you must choose an action to take for the tasks found.

To choose an action for a task:

1. Select the check box next to the task for which you want to choose an action.
2. Perform one of the following:
   - To remove the password in the task properties, click **Delete credentials**.

     The task is switched to run under the default account.
   - To replace the password with a new one, click **Enforce the password change even if the old password is wrong or not provided**.
   - To cancel the password change, click **No action is selected**.

The chosen actions are applied after you move to the next step of the wizard.

Step 3. Viewing the results

On the last step of the wizard, view the results for each of the found tasks. To complete the wizard, click the **Finish** button.

Viewing task run results stored on the Administration Server

Open Single Management Platform allows you to view the results for group tasks, tasks for specific devices, and Administration Server tasks.

To view the task results:

1. In the task properties window, select the **General** section.
2. Click the **Results** link to open the **Task results** window.

To view task results for a secondary Administration Server:

1. In the task properties window, select the **General** section.
2. Click the **Results** link to open the **Task results** window.
3. Click **Statistics from secondary Servers**.
4. Select the secondary Server for which you want to display the **Task results** window.

Manual setup of the group task for scanning a device with Kaspersky Endpoint Security

If the automatically specified schedule of the group scanning task is not appropriate for your organization, you must manually set up the most convenient schedule for this task based on the workplace rules adopted in the organization.

For example, the task is assigned a **Run on Fridays at 7:00 PM** schedule with automatic randomization, and the **Run missed tasks** check box is cleared. This means that if the devices in the organization are shut down on Fridays, for example, at 6:30 PM, the device scan task will never run. In this case you need to set up the group scanning task manually.

Managing detection and response tasks

When managing the program console, you can use tasks to perform various operations. For example, you can import assets or export KUMA event information to a TSV file.

Viewing the tasks table

The task table contains a list of created tasks and is located in the **Asset management** → **Tasks** → **Detection and response** section of the program console window. You can view the tasks that were created by you (current user).

A user with the Main administrator role can view the tasks of all users.

The tasks table contains the following information:

- **State**—the state of the task. One of the following statuses can be assigned to a task:
  - *Green dot blinking*—the task is active.
  - **Completed**—the task is complete.
  - **Cancel**—the task was canceled by the user.
  - **Error**—the task was not completed because of an error. The error message is displayed if you hover the mouse over the exclamation mark icon.
- **Task**—the task type. The program provides the following types of tasks:
  - **Events export**—export events.
  - **Run query in the background**—runs an SQL query in the background.
  - **Threat Lookup**—request data from the Kaspersky Threat Intelligence Portal.
  - **Retroscan**—task for replaying events.
  - **OSMP assets import**—imports asset data from Kaspersky Security Center Servers.

- **Accounts import**—imports user data from Active Directory.
- **KICS for Networks assets import**—imports asset data from KICS for Networks.
- **Repository update**—updates the KUMA repository to receive the resource packages from the source specified in settings.
- **Partition backup**—backs up partitions and saves the backup file at a specified path.
- **Restoring partitions from backup**—restores partitions from a backup copy.
- **Importing partitions**—imports partitions from an external KUMA installation.

- **Created by**—the user who created the task. If the task was created automatically, the column will show **Scheduled task**.

  This column is displayed only for users with the Main administrator and Tenant administrator roles.
- **Created**—task creation time.
- **Updated**—time when the task was last updated.
- **Tenant**—the name of the tenant in which the task was started.

The task date format depends on the localization language selected in the application settings. Possible date format options:

- English localization: YYYY-MM-DD.
- Russian localization: DD.MM.YYYY.

Viewing task run results

To view the results of a task:

1. In the main menu, go to **Asset management** → **Tasks** → **Detection and response**.

   The tasks table is displayed.
2. Click the link containing the task type in the **Task** column.

   A list of the operations available for this task type will be displayed.
3. Select **Show results**.

The task results window opens.

Restarting a task

To restart a task:

1. In the main menu, go to **Asset management** → **Tasks** → **Detection and response**.

   The tasks table is displayed.

2. Select the check box next to the task that you want to restart.

3. Click the **Restart** button.

   The task will be restarted.

Application tags

Open Single Management Platform enables you to tag the applications from applications registry. A tag is the label of an application that can be used for grouping or finding applications. A tag assigned to applications can serve as a condition in device selections.

For example, you can create the `[Browsers]` tag and assign it to all browsers such as Microsoft Internet Explorer, Google Chrome, Mozilla Firefox.

Creating an application tag

To create an application tag:

1. In the main menu, go to **Operations** → **Third-party applications** → **Application tags**.

2. Click **Add**.

   A new tag window opens.

3. Enter the tag name.

4. Click **OK** to save the changes.

   The new tag appears in the list of application tags.

Renaming an application tag

To rename an application tag:

1. In the main menu, go to **Operations** → **Third-party applications** → **Application tags**.

2. Select the check box next to the tag that you want to rename, and then click **Edit**.

   A tag properties window opens.

3. Change the tag name.

4. Click **OK** to save the changes.

   The updated tag appears in the list of application tags.

Assigning tags to an application

To assign one or several tags to an application:

1. In the main menu, go to **Operations** → **Third-party applications** → **Applications registry**.
2. Click the name of the application to which you want to assign tags.
3. Select the **Tags** tab.

   The tab displays all application tags that exist on the Administration Server. For tags assigned to the selected application, the check box in the **Tag assigned** column is selected.
4. For tags that you want to assign, select check boxes in the **Tag assigned** column.
5. Click **Save** to save the changes.

The tags are assigned to the application.

Removing assigned tags from an application

To remove one or several tags from an application:

1. In the main menu, go to **Operations** → **Third-party applications** → **Applications registry**.
2. Click the name of the application from which you want to remove tags.
3. Select the **Tags** tab.

   The tab displays all application tags that exist on the Administration Server. For tags assigned to the selected application, the check box in the **Tag assigned** column is selected.
4. For tags that you want to remove, clear check boxes in the **Tag assigned** column.
5. Click **Save** to save the changes.

The tags are removed from the application.

> The removed application tags are not deleted. If you want, you can delete them manually.

Deleting an application tag

To delete an application tag:

1. In the main menu, go to **Operations** → **Third-party applications** → **Application tags**.
2. In the list, select the application tag that you want to delete.

3. Click the **Delete** button.

4. In the window that opens, click **OK**.

   The application tag is deleted. The deleted tag is automatically removed from all of the applications to which it was assigned.

Granting offline access to the external device blocked by Device Control

In Device Control component of the Kaspersky Endpoint Security policy, you can manage user access to external devices that are installed on or connected to the client device (for example, hard drives, cameras, or Wi-Fi modules). This lets you protect the client device from infection when such external devices are connected, and prevent loss or leaks of data.

If you need to grant temporary access to the external device blocked by Device Control, but it is not possible to add the device to the list of trusted devices, you can grant temporary offline access to the external device. Offline access means that the client device has no access to the network.

> You can grant offline access to the external device blocked by Device Control only if the **Allow requests for temporary access** option is enabled in the settings of the Kaspersky Endpoint Security policy, in the **Application settings** → **Security Controls** → **Device Control** section.

Granting offline access to the external device blocked by Device Control includes the following stages:

1. In the Kaspersky Endpoint Security dialog window, device user who wants to have access to the blocked external device, generates a request access file and sends it to the Open Single Management Platform administrator.

2. Getting this request, the Open Single Management Platform administrator creates an access key file and send it to the device user.

3. In the Kaspersky Endpoint Security dialog window, the device user activates the access key file and obtains temporary access to the external device.

To grant temporary access to the external device blocked by Device Control:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The list of managed devices is displayed.

2. In this list, select the user's device that requests access to the external device blocked by Device Control.

   You can select only one device.

3. Above the list of managed devices, click the ellipsis button (**...**), and then click the **Grant access to the device in offline mode** button.

4. In the **Application settings** window that opens, in the **Device Control** section, click the **Browse** button.

5. Select the request access file that you have received from the user, and then click the **Open** button. The file should have the AKEY format.

   The details of the locked device to which the user has requested access is displayed.

6. Specify the value of the **Access duration** setting.

   This setting defines the length of time for which you grant the user access to the locked device. The default value is the value that was specified by the user when creating the request access file.

7. Specify the time period during which the access key can be activated on the device.

   This setting defines the time period during which the user can activate access to the blocked device by using the provided access key.

8. Click the **Save** button.

9. In the window that opens, select the destination folder in which you want to save the file containing the access key for the blocked device.

10. Click the **Save** button.

As a result, when you send the user the access key file and the user activates it in the Kaspersky Endpoint Security dialog window, the user has temporary access to the blocked device for the specific period.

Registering Kaspersky Industrial CyberSecurity for Networks application in OSMP Console

To start working with the Kaspersky Industrial CyberSecurity for Networks application via OSMP Console, you must first register it in OSMP Console.

To register the Kaspersky Industrial CyberSecurity for Networks application:

1. Make sure that the following is done:

   - You have downloaded and installed the Kaspersky Industrial CyberSecurity for Networks web plug-in.

     You can do it later while waiting for the Kaspersky Industrial CyberSecurity for Networks Server to synchronize with the Administration Server. After the plug-in is downloaded and installed, the **KICS for Networks** section is displayed in the OSMP Console main menu.

   - In the Kaspersky Industrial CyberSecurity for Networks web interface, interaction with Open Single Management Platform is configured and enabled. For details, refer to the Kaspersky Industrial CyberSecurity for Networks Online Help.

2. Move the device where Kaspersky Industrial CyberSecurity for Networks Server is installed from the Unassigned devices group to the Managed devices group:

   a. In the main menu, go to **Discovery & deployment** → **Unassigned devices**.

   b. Select the check box next to the device where the Kaspersky Industrial CyberSecurity for Networks Server is installed.

   c. Click the **Move to group** button.

   d. In the hierarchy of administration groups, select the check box next to the **Managed devices** group.

   e. Click the **Move** button.

3. Open the properties window of the device where the Kaspersky Industrial CyberSecurity for Networks Server is installed.

4. On the device properties page, in the **General** section, select the **Do not disconnect from the Administration Server** option, and then click the **Save** button.

5. On the device properties page, select the **Applications** section.

6. In the **Applications** section, select Kaspersky Security Center Network Agent.

7. If the current status of the application is *Stopped*, wait until it changes to *Running*.

   This may take up to 15 minutes. If you have not yet installed the Kaspersky Industrial CyberSecurity for Networks web plug-in, you can do it now.

8. If you want to view the statistics of Kaspersky Industrial CyberSecurity for Networks, you may add widgets on the dashboard. To add the widgets, do the following:

   a. In the main menu, go to **Monitoring** → **Dashboard**.

   b. On the dashboard, click the **Add or restore web widget** button.

   c. In the widget menu that opens, select **Other**.

   d. Select the widgets that you want to add:

      - KICS for Networks deployment map
      - Information about KICS for Networks Servers
      - Up-to-date events of KICS for Networks
      - Devices with issues in KICS for Networks
      - Critical events in KICS for Networks
      - Statuses in KICS for Networks

9. To proceed to the Kaspersky Industrial CyberSecurity for Networks web interface, do the following:

   a. In the main menu, go to **KICS for Networks** → **Search**.

   b. Click the **Find events or devices** button.

   c. In the **Query parameters** window that opens, click the **Server** field.

   d. Select the Kaspersky Industrial CyberSecurity for Networks Server from the drop-down list of servers that are integrated with Open Single Management Platform, and then click the **Find** button.

   e. Click the **Go to Server** link next to the name of the Kaspersky Industrial CyberSecurity for Networks Server.

      The Kaspersky Industrial CyberSecurity for Networks sign-in page is displayed.

      > To log in to the Kaspersky Industrial CyberSecurity for Networks web interface, you need to provide the application user account credentials.

Managing users and user roles

This section describes users and user roles, and provides instructions for creating and modifying them, for assigning roles and groups to users, and for associating policy profiles with roles.

About user accounts

Open Single Management Platform allows you to manage user accounts and security groups. The application supports four types of accounts:

- Accounts of organization employees. Administration Server retrieves data of the accounts of those domain users when polling the organization's domain controller.
- Accounts of local users. Local accounts on managed devices, as well as the local accounts on the device on which Administration Server is installed.
- Accounts of internal users of Open Single Management Platform. You can create accounts of internal users. These accounts are used only within Open Single Management Platform.
- Service accounts. These accounts are only used to connect Kaspersky Next XDR Expert services to Administration Server. Service accounts are added automatically when the solution is deployed, and cannot be deleted or changed. The rights and roles of these accounts cannot be changed, either. You cannot use a service account to sign in to OSMP Console.

The kladmins group cannot be used to access OSMP Console in Open Single Management Platform. The kladmins group can only contain accounts that are used to start Open Single Management Platform services.

The default authentication method in OSMP Console is to enter user name, password, and an additional single-use security code for two-factor authentication. Starting with Kaspersky Next XDR Expert 2.0, two-factor authentication is enabled automatically and the option to configure two-factor authentication at sign-in is disabled by default.

You can configure domain authentication over the Kerberos protocol. Domain authentication allows you to enable secure authentication in OSMP Console without having to re-enter the password on the corporate network (single sign-on).

To view tables of user accounts and security groups:

1. In the main menu, go to **Users & roles** → **Users & groups**.
2. Select the **Users** or the **Groups** tab.

The table of users or security groups opens.

If you want to view the table with only internal, or local, or domain users or groups, set the **Subtype** filter criteria to the corresponding value.

If the table of users contains more than 50,000 users, only the first 50,000 users are displayed, and the remaining users are not shown.

About user roles

A *user role* (also referred to as a *role*) is an object containing a set of rights and privileges. A role can be associated with settings of Kaspersky applications installed on a user device. You can assign a role to a set of users or to a set of security groups at any level in the hierarchy of administration groups, Administration Servers, or at the level of specific objects.

If you manage devices through a hierarchy of Administration Servers that includes virtual Administration Servers, note that you can create, modify, or delete user roles only from a physical Administration Server. Then, you can propagate the user roles to secondary Administration Servers, including virtual ones.

You can associate user roles with policy profiles. If a user is assigned a role, this user gets security settings necessary to perform job functions.

A user role can be associated with users of devices in a specific administration group.

User role scope

A *user role scope* is a combination of users and administration groups. Settings associated with a user role apply only to devices that belong to users who have this role, and only if these devices belong to groups associated with this role, including child groups.

Advantage of using roles

An advantage of using roles is that you do not have to specify security settings for each of the managed devices or for each of the users separately. The number of users and devices in a company may be quite large, but the number of different job functions that require different security settings is considerably smaller.

Differences from using policy profiles

Policy profiles are properties of a policy that is created for each Kaspersky application separately. A role is associated with many policy profiles created for different applications. Therefore, a role is a method of uniting settings for a certain user type in one place.

Configuring access rights to application features. Role-based access control

Open Single Management Platform provides facilities for role-based access to the features of Open Single Management Platform and managed Kaspersky applications.

You can configure access rights to application features for Open Single Management Platform users in one of the following ways:

- By configuring the rights for each user or group of users individually.
- By creating standard user roles with a predefined set of rights and assigning those roles to users depending on their scope of duties.

Application of user roles is intended to simplify and shorten routine procedures of configuring users' access rights to application features. Access rights within a role are configured in accordance with the standard tasks and the users' scope of duties.

User roles can be assigned names that correspond to their respective purposes. You can create an unlimited number of roles in the application.

You can use the predefined user roles with already configured set of rights, or create new roles and configure the required rights yourself.

Access rights to application features

The table below shows the Open Single Management Platform features with the access rights to manage the associated tasks, reports, settings, and perform the associated user actions.

To perform the user actions listed in the table, a user has to have the right specified next to the action.

Read, **Write**, and **Execute** rights are applicable to any task, report, or setting. In addition to these rights, a user has to have the **Perform operations on device selections** right to manage tasks, reports, or settings on device selections.

All tasks, reports, settings, and installation packages that are missing in the table belong to the **General features: Basic functionality** functional area.

Access rights to application features

Functional area	Right	User action: right required to perform the action	Task	Report	Other
General features: Management of administration groups	**Write**	• Add device to an administration group: **Write** • Delete device from an administration group: **Write** • Add an administration group to another administration group: **Write** • Delete an administration group from another administration group: **Write**	None	None	None

Functional area	Right	User action: right required to perform the action	Task	Report	Other
General features: Access objects regardless of their ACLs	**Read**	Get read access to all objects: **Read**	None	None	The **General features: Access objects regardless of their ACLs** functional area is intended for audit purposes. When users are granted **Read** rights in this functional area, they get full **Read** access to all objects and are able to execute any created tasks on selections of devices connected to the Administration Server via Network Agent with local administrator rights (root for Linux). We recommend granting these rights carefully and to a limited set of users who need them to perform their official duties. Access is granted regardless of other rights, even if they prohibit read access to specific objects.
General features: Basic functionality	• **Read** • **Write** • **Execute** • **Perform operations on device selections**	• Device moving rules (create, modify, or delete) for the virtual Server: **Write**, **Perform operations on device selections** • Get Mobile (LWNGT) protocol custom certificate: **Read** • Set Mobile (LWNGT) protocol custom certificate: **Write** • Get NLA-defined network list: **Read** • Add, modify, or delete NLA-defined network list: **Write** • View Access Control List of groups: **Read** • View the operating system log: **Read**	• "Download updates to the Administration Server repository" • "Deliver reports" • "Distribute installation package" • "Install application on secondary Administration Servers remotely"	• "Report on protection status" • "Report on threats" • "Report on most heavily infected devices" • "Report on status of anti-virus databases" • "Report on errors" • "Report on network attacks" • "Summary report on perimeter defense applications installed" • "Summary report on types of applications installed" • "Report on users of infected devices" • "Report on incidents" • "Report on events" • "Report on activity of distribution points" • "Report on secondary Administration Servers" • "Report on Device Control events" • "Report on vulnerabilities"	None

Functional area	Right	User action: right required to perform the action	Task	Report	Other
				• "Report on prohibited applications" • "Report on Web Control" • "Report on encryption status of managed devices" • "Report on encryption status of mass storage devices" • "Report on rights to access encrypted drives" • "Report on file encryption errors" • "Report on blockage of access to encrypted files" • "Report on effective user permissions" • "Report on rights"	
General features: **Deleted objects**	• **Read** • **Write**	• View deleted objects in the Recycle Bin: **Read** • Delete objects from the Recycle Bin: **Write**	None	None	None
General features: **Event processing**	• **Delete events** • **Edit event notification settings** • **Edit event logging settings** • **Write**	• Change events registration settings: **Edit event logging settings** • Change events notification settings: **Edit event notification settings** • Delete events: **Delete events**	None	None	Settings: • The maximum number of events stored in the database • Period for storing events from the deleted devices
General features: **Operations on Administration Server**	• **Read** • **Write** • **Execute** • **Modify object ACLs** • **Perform operations on device selections**	• Specify ports of Administration Server for the network agent connection: **Write** • Specify ports of Activation Proxy launched on the Administration Server: **Write** • Specify ports of Activation Proxy for Mobile launched on the Administration Server: **Write** • Specify ports of the Web Server for distribution of standalone packages: **Write**	• "Backup of Administration Server data" • "Databases maintenance"	None	None

Functional area	Right	User action: right required to perform the action	Task	Report	Other
		• Specify ports of the Web Server for distribution of MDM profiles: **Write** • Specify SSL-ports of the Administration Server for connection via OSMP Console: **Write** • Specify ports of the Administration Server for mobile connection: **Write** • Specify the maximum number of events stored in the Administration Server database: **Write** • Specify the maximum number of events that can be sent by the Administration Server: **Write** • Specify time period during which events can be sent by the Administration Server: **Write**			
General features: **Kaspersky software deployment**	• **Manage Kaspersky patches** • **Read** • **Write** • **Execute** • **Perform operations on device selections**	Approve or decline installation of the patch: **Manage Kaspersky patches**	None	• "Report on license key usage by virtual Administration Server" • "Report on Kaspersky software versions" • "Report on incompatible applications" • "Report on versions of Kaspersky software module updates" • "Report on protection deployment"	Installation package: "Kaspersky"
General features: **Key management**	• **Export key file** • **Write**	• Export key file: **Export key file** • Modify Administration Server license key settings: **Write**	None	None	None
General features: **Enforced report management**	• **Read** • **Write**	• Create reports regardless of their ACLs: **Write** • Execute reports regardless of their ACLs: **Read**	None	None	None
General features: **Hierarchy of**	**Configure hierarchy of**	• Register, update, or delete secondary	None	None	None

Functional area	Right	User action: right required to perform the action	Task	Report	Other
Administration Servers	**Administration Servers**	Administration Servers: **Configure hierarchy of Administration Servers**			
General features: **User permissions**	**Modify object ACLs**	• Change Security properties of any object: **Modify object ACLs** • Manage user roles: **Modify object ACLs** • Manage internal users: **Modify object ACLs** • Manage security groups: **Modiobject ACLs** • Manage aliases: **Modify object ACLs**	None	None	None
General features: **Virtual Administration Servers**	• **Manage virtual Administration Servers** • **Read** • **Write** • **Execute** • **Perform operations on device selections**	• Get list of virtual Administration Servers: **Read** • Get information on the virtual Administration Server: **Read** • Create, update, or delete a virtual Administration Server: **Manage virtual Administration Servers** • Move a virtual Administration Server to another group: **Manage virtual Administration Servers** • Set administration virtual Server permissions: **Manage virtual Administration Servers**	None	None	None
General features: **Encryption Key Management**	**Write**	Approval of the launch of the playbook in training mode the encryption keys: **Write**	None	None	None
System management: **Vulnerability and patch management**	• **Read** • **Write** • **Execute** • **Perform operations on device selections**	• View third-party patch properties: **Read** • Change third-party patch properties: **Write**	• "Fix vulnerabilities" • "Install required updates and fix vulnerabilities"	"Report on software updates"	None

Predefined user roles

User roles assigned to Open Single Management Platform users provide them with sets of access rights to application features.

You can use the predefined user roles with already configured set of rights, or create new roles and configure the required rights yourself. Some of the predefined user roles available in Open Single Management Platform can be associated with specific job positions, for example, **Auditor**, **Security Officer**, **Supervisor**. Access rights of these roles are pre-configured in accordance with the standard tasks and scope of duties of the associated positions. The table below shows how roles can be associated with specific job positions.

Examples of roles for specific job positions

Role	Description
Auditor	Permits all operations with all types of reports, all viewing operations, including viewing deleted objects (grants the **Read** and **Write** permissions in the **Deleted objects** area). Does not permit other operations. You can assign this role to a person who performs the audit of your organization.
Supervisor	Permits all viewing operations; does not permit other operations. You can assign this role to a security officer and other managers in charge of the IT security in your organization.
Security Officer	Permits all viewing operations, permits reports management; grants limited permissions in the **System management**: **Connectivity** area. You can assign this role to an officer in charge of the IT security in your organization.

The table below shows the access rights assigned to each predefined user role.

Access rights of predefined user roles

Role	Description
Basic roles	

Administration Server Administrator	Permits all operations in the following functional areas:
	• **General features**:
	  • **Basic functionality**
	  • **Encryption key management**
	  • **Event processing**
	  • **Hierarchy of Administration Servers**
	  • **Virtual Administration Servers**
	• **System management**:
	  • **Connectivity**
	  • **Hardware inventory**
	  • **Software inventory**

Administration Server Operator	Grants the **Read** and **Execute** rights in all of the following functional areas:
	• **General features**:
	  • **Basic functionality**
	  • **Encryption key management**
	  • **Virtual Administration Servers**
	• **System management**:
	  • **Connectivity**
	  • **Hardware inventory**
	  • **Software inventory**

Auditor	Permits all operations in the following functional areas, in **General features**: • **Access objects regardless of their ACLs** • **Deleted objects** • **Enforced report management** You can assign this role to a person who performs the audit of your organization.
Installation Administrator	Permits all operations in the following functional areas, in **General features**: • **Basic functionality** • **Kaspersky software deployment** • **License key management** Grants **Read** and **Execute** rights in the **General features: Virtual Administration Servers** functional area.

Role	Description
Installation Operator	Grants the **Read** and **Execute** rights in all of the following functional areas, in **General features**: • **Basic functionality** • **Kaspersky software deployment** (also grants the **Manage Kaspersky Lab patches** right in this area) • **Virtual Administration Servers**
Kaspersky Endpoint Security Administrator	Permits all operations in the following functional areas: • **General features: Basic functionality** • Kaspersky Endpoint Security area, including all features Grants the **Read** and **Write** rights in the **General features: Encryption key management** functional area.
Kaspersky Endpoint Security Operator	Grants the **Read** and **Execute** rights in all of the following functional areas: • **General features: Basic functionality** • Kaspersky Endpoint Security area, including all features
Main Administrator	Permits all operations in functional areas, *except* for the following areas, in **General features**: • **Access objects regardless of their ACLs** • **Enforced report management** Grants the **Read** and **Write** rights in the **General features: Encryption key management** functional area.
Main Operator	Grants the **Read** and **Execute** (where applicable) rights in all of the following functional areas: • **General features**: • **Basic functionality** • **Deleted objects** • **Operations on Administration Server** • **Kaspersky Lab software deployment** • **Virtual Administration Servers** • Kaspersky Endpoint Security area, including all features
Mobile Device Management Administrator	Permits all operations in the **General features: Basic functionality** functional area.
Security Officer	Permits all operations in the following functional areas, in **General features**: • **Access objects regardless of their ACLs** • **Enforced report management** Grants the **Read**, **Write**, **Execute**, **Save files from devices to the administrator's workstation**, and **Perform operations on device selections** rights in the **System management: Connectivity** functional area. You can assign this role to an officer in charge of the IT security in your organization.
Self Service Portal User	Permits all operations in the **Mobile Device Management: Self Service Portal** functional area. This feature is not supported in Kaspersky Security Center 11 and later version.
Supervisor	Grants the **Read** right in the **General features**: **Access objects regardless of their ACLs** and **General features**: **Enforced report management** functional areas. You can assign this role to a security officer and other managers in charge of the IT security in your organization.
XDR roles	
Main administrator	Permits all operations in the XDR functional areas: • **Alerts and incidents** • **Playbooks and response** • **Asset Management** • **IAM** • **Tenants** • **Integrations** • **Licenses**
Tenant administrator	Permits all operations in the XDR functional areas: • **Alerts and incidents** • **Playbooks and response** • **Asset Management** • **IAM** • **Tenants** • **Integrations** • **Licenses** This role corresponds to the Main Administrator role, but it has a restriction. In KUMA, a tenant administrator has limited access to the preset objects.

Role	Description
SOC administrator	Grants the following rights in the XDR functional areas: • **Playbooks and response**: **Read**, **Write**, and **Delete** • **IAM**: **Read users and roles**, **Assign roles**, and **Lists users** • **Tenants**: **Read** and **Write** • **Integrations**: **Read**, **Write**, and **Delete** • **Licenses**: **Read** • **Asset Management**

Role	Description
Junior analyst	Grants the following rights in the XDR functional areas: • **Alerts and incidents**: **Read** and **Write** • **Asset Management**: **Read** **Viewing the details** Read: • View the list of categories. • Add the list of devices in terms of categories. • Export devices in CSV format. • View devices that meet the criteria of active categorization. • View an asset card. • View the list of custom fields. • View TTL for the archived devices. • View assets audit settings. • View the integration with Open Single Management Platform settings. • View the threshold settings to define the activity status of a device. Write: • Add categories. • Modify categories. • Add a device. • Modify a device, including the category. • Run forced synchronization of Open Single Management Platform devices. • Run active categorization manually. • Add asset custom fields. • Modify asset custom fields. • Modify TTL for the archived devices. • Add assets audit settings. • Modify assets audit settings. • Duplicate assets audit settings. • Add the integration with Open Single Management Platform settings. • Modify the integration with Open Single Management Platform settings. • Modify the threshold settings to define the activity status of a device. Delete: • Delete categories. • Delete a device. • Delete asset custom fields. • Delete assets audit settings.

Role	Description
	 - **IAM**: **Read users and roles** and **Lists users** - **Tenants**: **Read** - **Integrations**: **Read** - **Licenses**: **Read**
Tier 2 analyst	Grants the following rights in the XDR functional areas: - **Alerts and incidents**: **Read** and **Write** - **Playbooks and response**: **Read**, **Write**, **Delete**, and **Execute** - **Asset Management** - **IAM**: **Read users and roles** and **Lists users** - **Tenants**: **Read** - **Integrations**: **Read** - **Licenses**: **Read**
Tier 1 analyst	Grants the following rights in the XDR functional areas: - **Alerts and incidents**: **Read** and **Write** - **Playbooks and response**: **Read**, **Write**, **Delete**, and **Execute** - **Asset Management** - **IAM**: **Read users and roles** and **Lists users** - **Tenants**: **Read** - **Integrations**: **Read** - **Licenses**: **Read** This role corresponds to the Tier 2 analyst role, but it has a restriction. In KUMA, a Tier 1 analyst can only modify their own objects.

Role	Description
SOC manager	Grants the following rights in the XDR functional areas: • **Alerts and incidents**: **Read** and **Write** • **Playbooks and response**: **Read** • **Asset Management**: **Read** **Viewing the details** Read: • View the list of categories. • Add the list of devices in terms of categories. • Export devices in CSV format. • View devices that meet the criteria of active categorization. • View an asset card. • View the list of custom fields. • View TTL for the archived devices. • View assets audit settings. • View the integration with Open Single Management Platform settings. • View the threshold settings to define the activity status of a device. Write: • Add categories. • Modify categories. • Add a device. • Modify a device, including the category. • Run forced synchronization of Open Single Management Platform devices. • Run active categorization manually. • Add asset custom fields. • Modify asset custom fields. • Modify TTL for the archived devices. • Add assets audit settings. • Modify assets audit settings. • Duplicate assets audit settings. • Add the integration with Open Single Management Platform settings. • Modify the integration with Open Single Management Platform settings. • Modify the threshold settings to define the activity status of a device. Delete: • Delete categories. • Delete a device. • Delete asset custom fields. • Delete assets audit settings.

Role	Description
	 • IAM: Read users and roles and Lists users • Tenants: Read • Integrations: Read • Licenses: Read

Role	Description
Approver	Grants the following rights in the XDR functional areas: • **Alerts and incidents**: **Read, Write, Close** • **Playbooks and response**: **Read** and **Response confirmation** • **Asset Management**: **Read** **Viewing the details** Read: • View the list of categories. • Add the list of devices in terms of categories. • Export devices in CSV format. • View devices that meet the criteria of active categorization. • View an asset card. • View the list of custom fields. • View TTL for the archived devices. • View assets audit settings. • View the integration with Open Single Management Platform settings. • View the threshold settings to define the activity status of a device. Write: • Add categories. • Modify categories. • Add a device. • Modify a device, including the category. • Run forced synchronization of Open Single Management Platform devices. • Run active categorization manually. • Add asset custom fields. • Modify asset custom fields. • Modify TTL for the archived devices. • Add assets audit settings. • Modify assets audit settings. • Duplicate assets audit settings. • Add the integration with Open Single Management Platform settings. • Modify the integration with Open Single Management Platform settings. • Modify the threshold settings to define the activity status of a device. Delete: • Delete categories. • Delete a device. • Delete asset custom fields. • Delete assets audit settings.

Role	Description
	 • IAM: Read users and roles • Tenants: Read • Integrations: Read • Licenses: Read

Role	Description
Auditor	Grants the following rights in the XDR functional areas: • **Alerts and incidents**: **Read** • **Playbooks and response**: **Read** • **Asset Management**: **Read** **Viewing the details** Read: • View the list of categories. • Add the list of devices in terms of categories. • Export devices in CSV format. • View devices that meet the criteria of active categorization. • View an asset card. • View the list of custom fields. • View TTL for the archived devices. • View assets audit settings. • View the integration with Open Single Management Platform settings. • View the threshold settings to define the activity status of a device. Write: • Add categories. • Modify categories. • Add a device. • Modify a device, including the category. • Run forced synchronization of Open Single Management Platform devices. • Run active categorization manually. • Add asset custom fields. • Modify asset custom fields. • Modify TTL for the archived devices. • Add assets audit settings. • Modify assets audit settings. • Duplicate assets audit settings. • Add the integration with Open Single Management Platform settings. • Modify the integration with Open Single Management Platform settings. • Modify the threshold settings to define the activity status of a device. Delete: • Delete categories. • Delete a device. • Delete asset custom fields. • Delete assets audit settings.

Role	Description
	• **IAM**: **Read users and roles** and **Lists users** • **Tenants**: **Read** • **Integrations**: **Read** • **Licenses**: **Read**
Interaction with child incidents	Grants the following rights in the XDR functional areas: • **Child incidents**: **Read** and **Write** • **IAM**: **Read users and roles** In addition to this role, you must have one of the following XDR roles: Main administrator, Tenant administrator, Junior analyst, Tier 2 analyst, Tier 1 analyst, SOC manager, Approver, Auditor.
Manage shared resources	Grants the right to edit access to shared resources located in the shared tenant.
Read shared resources	Grants the right to read access to shared resources located in the shared tenant.
Service roles	
Automatic Threat Responder	Grants service accounts the right to respond to threats. Access rights are configured automatically in accordance with the role-based access control policies of Kaspersky Security Center Linux and managed Kaspersky applications. You can assign this role only to service accounts. This role cannot be edited.

Assigning access rights to specific objects

In addition to assigning access rights at the server level, you can configure access to specific objects, for example, to a specific task. The application allows you to specify access rights to the following object types:

- Administration groups
- Tasks
- Reports
- Device selections
- Event selections

To assign access rights to a specific object:

1. Depending on the object type, in the main menu, go to the corresponding section:
   - **Asset management → Hierarchy of groups**
   - **Asset management → Tasks**
   - **Monitoring → Reports**

- **Asset management** → **Device selections**
- **Monitoring** → **Event selections**

2. Open the properties of the object to which you want to configure access rights.

   To open the properties window of an administration group or a task, click the object name. Properties of other objects can be opened by using the button on the toolbar.

3. In the properties window, open the **Access rights** section.

   The user list opens. The listed users and security groups have access rights to the object. By default, if you use a hierarchy of administration groups or Servers, the list and access rights are inherited from the parent administration group or primary Server.

4. To be able to modify the list, enable the **Use custom permissions** option.

5. Configure access rights:

   - Use the **Add** and **Delete** buttons to modify the list.
   - Specify access rights for a user or security group. Do one of the following:
     - If you want to specify access rights manually, select the user or security group, click the **Access rights** button, and then specify the access rights.
     - If you want to assign a user role to the user or security group, select the user or security group, click the **Roles** button, and then select the role to assign.

6. Click the **Save** button.

The access rights to the object are configured.

Assigning permissions to users and groups

You can give users and security groups access rights to use different features of Administration Server and of the Kaspersky applications for which you have management plug-ins, for example, Kaspersky Endpoint Security for Windows.

To assign permissions to a user or security group:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **Access rights** tab, select the check box next to the name of the user or the security group to whom to assign rights, and then click the **Access rights** button.

   You cannot select multiple users or security groups at the same time. If you select more than one item, the **Access rights** button will be disabled.

3. Configure the set of rights for the user or group:

   a. Expand the node with features of Administration Server or other Kaspersky application.

   b. Select the **Allow** or **Deny** check box next to the feature or the access right that you want.

Example 1: Select the **Allow** check box next to the **Application integration** node to grant all available access rights to the Application integration feature (**Read**, **Write**, and **Execute**) for a user or group.

Example 2: Expand the **Encryption key management** node, and then select the **Allow** check box next to the **Write** permission to grant the **Write** access right to the Encryption key management feature for a user or group.

4. After you configure the set of access rights, click **OK**.

The set of rights for the user or group of users will be configured.

The permissions of the Administration Server (or the administration group) are divided into the following areas:

- General features:
  - Management of administration groups
  - Access objects regardless of their ACLs
  - Basic functionality
  - Deleted objects
  - Encryption Key Management
  - Event processing
  - Operations on Administration Server
  - Device tags
  - Kaspersky software deployment
  - License key management
  - Enforced report management
  - Hierarchy of Servers
  - User rights
  - Virtual Administration Servers
- Mobile Device Management:
  - General
- System Management:
  - Connectivity
  - Hardware inventory
  - Network Access Control
  - Deploy operating system

- Manage vulnerabilities and patches
- Remote installation
- Software inventory

If neither **Allow** nor **Deny** is selected for a permission, then the permission is considered *undefined*: it is denied until it is explicitly denied or allowed for the user.

The rights of a user are the sum of the following:

- User's own rights
- Rights of all the roles assigned to this user
- Rights of all the security group to which the user belongs
- Rights of all the roles assigned to the security groups to which the user belongs

If at least one of these sets of rights has **Deny** for a permission, then the user is denied this permission, even if other sets allow it or leave it undefined.

You can also add users and security groups to the scope of a user role to use different features of Administration Server. Settings associated with a user role will only apply to devices that belong to users who have this role, and only if these devices belong to groups associated with this role, including child groups.

Adding an account of an internal user

To add a new internal user account to Open Single Management Platform:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.
2. Click **Add**.
3. In the **Add user** window that opens, specify the settings of the new user account:
   - **Name**.
   - **Password** for the user connection to Open Single Management Platform.

     The password must comply with the following rules:
     - The password must be 8 to 256 characters long.
     - The password must contain characters from at least three of the groups listed below:
       - Uppercase letters (A-Z)
       - Lowercase letters (a-z)

- Numbers (0-9)
- Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)

- The password must not contain any whitespaces, Unicode characters, or the combination of "." and "@", when "." is placed before "@".

To see the characters that you entered, click and hold the **Show** button.

> The number of attempts for entering the password is limited. By default, the maximum number of allowed password entry attempts is 5. This number is fixed and cannot be changed.

> If the user enters an invalid password the specified number of times, the user account is blocked for one hour. You can unblock the user account only by changing the password.

4. Click **Save** to save the changes.

A new user account is added to the user list.

> To sign in with a created internal user account on the OSMP Console, you must assign a role to the account. Before signing in, make sure that the internal user account has a role. Otherwise, a login error will occur.

Creating a security group

To create a security group:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Groups** tab.
2. Click **Add**.
3. In the **Create security group** window that opens, specify the following settings for the new security group:
   - **Group name**
   - **Description**
4. Click **Save** to save the changes.

A new security group is added to the group list.

Editing an account of an internal user

To edit an internal user account in Open Single Management Platform:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.

2. Click the name of the user account that you want to edit.

3. In the user settings window that opens, on the **General** tab, change the settings of the user account:

   - **Description**

   - **Full name**

   - **Email address**

   - **Main phone**

   - **Set new password** for the user connection to Open Single Management Platform.

     The password must comply with the following rules:

     - The password must be 8 to 256 characters long.

     - The password must contain characters from at least three of the groups listed below:

       - Uppercase letters (A-Z)

       - Lowercase letters (a-z)

       - Numbers (0-9)

       - Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)

     - The password must not contain any whitespaces, Unicode characters, or the combination of "." and "@", when "." is placed before "@".

     To see the entered password, click and hold the **Show** button.

     > The number of attempts for entering the password is limited. By default, the maximum number of allowed password entry attempts is 5. This number is fixed and cannot be changed. If the user enters an invalid password the specified number of times, the user account is blocked for one hour. You can unblock the user account only by changing the password.

   - If necessary, switch the toggle button to **Disabled** to prohibit the user from connecting to the application. You can disable an account, for example, after an employee leaves the company.

4. On the **Two-factor authentication** tab, you can specify the security settings for this account.

5. On the **Groups** tab, you can add the user to security groups.

6. On the **Devices** tab, you can assign devices to the user.

7. On the **Roles** tab, you can assign roles to the user.

8. Click **Save** to save the changes.

   The updated user account appears in the list of users.

Changing a user work status

The user work status shows whether or not you can assign alerts and incidents to the user.

Users may have one of the following statuses:

- **Available**—The user is ready to work with alerts and incidents.
- **Busy**—The user is not ready to work with alerts and incidents. Alerts and incidents cannot be automatically assigned to the user.

By default, when a user signs in to OSMP Console or to KUMA Console, the user work status automatically updates to **Available**.

You can manually change your work status or the work status for the currently signed-in users.

Changing the work status for your own account

Your current work status is displayed in the main menu next to your account name as a green dot ( ) if the status is **Available**, or as a red dot ( ) if the status is **Busy**.

If you want to change your current work status, in the upper pane, go to your account settings, and then select the status to be assigned to your account.

Changing the work status for an internal user account

> The **Work status** field is displayed in the user settings window only if the user is currently signed in to OSMP Console or to KUMA Console.
> To change the work status for an internal user, you must have one of the following XDR roles: Main administrator, Tenant administrator, SOC administrator.

To change the work status of an internal user:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.

   A user list opens.

2. Click the name of the user account for which you want to change the status.

   The user settings window opens with the **General** tab selected.

3. In the **Work status** field, select the status that you want to assign: **Available** or **Busy**.

4. Click the **Save** button.

The window is closed, and the work status of the user is changed.

Editing a security group

To edit a security group:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Groups** tab.

2. Click the name of the security group that you want to edit.

3. In the group settings window that opens, change the settings of the security group:

   - On the **General** tab, you can change the **Name** and **Description** settings. These settings are available only for internal security groups.

   - On the **Users** tab, you can do the following:

     - View the table of the group users.

     - Add users to the security group. To do this, click the **Add** button, then in the **Add users** pane that opens, select the check boxes next to the user accounts that you want to add, and click **Save**.

       In the **Add users** pane, you can filter the users by tenant. To do this, set the **Tenant owner** filter criteria to the required value.

       > Adding users to the security group is available only for internal users and internal security groups.

     - Delete users from the security group. To do this, select the check box next to the user account name, click **Delete**, and then confirm the action.

   - On the **Roles** tab, you can assign a role to the security group.

4. Click **Save** to save the changes.

The changes are applied to the security group.

Assigning a role to a user or a security group

To assign a role to a user or a security group:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** or the **Groups** tab.

2. Select the name of the user or the security group to whom to assign a role.

   You can select multiple names.

3. On the menu line, click the **Assign role** button.

   The Role assignment wizard starts.

4. Follow the instructions of the wizard: select the role that you want to assign to the selected users or security groups, and then select the scope of role.

   A *user role scope* is a combination of users and administration groups. Settings associated with a user role apply only to devices that belong to users who have this role, and only if these devices belong to groups associated with this role, including child groups.

The role with a set of rights for working with Administration Server is assigned to the user (or users, or the security group). In the list of users or security groups, a check box appears in the **Has assigned roles** column.

Adding user accounts to an internal security group

You can add only accounts of internal users to an internal security group.

To add user accounts to an internal security group:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.

2. Select check boxes next to user accounts that you want to add to a security group.

3. Click the **Assign group** button.

4. In the **Assign group** window that opens, select the security group to which you want to add user accounts.

5. Click the **Save** button.

The user accounts are added to the security group. You can also add internal users to a security group by using the group settings.

Assigning a user as a device owner

For information about assigning a user as a mobile device owner, see Kaspersky Secure Mobility Management Help.

To assign a user as a device owner:

1. If you want to assign an owner of a device connected to a virtual Administration Server, first switch to the virtual Administration Server:

   a. In the main menu, click the chevron icon ( ) to the right of the current Administration Server name.

   b. Select the required Administration Server.

2. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.

   A user list opens. If you are currently connected to a virtual Administration Server, the list includes users from the current virtual Administration Server and the primary Administration Server.

3. Click the name of the user account that you want to assign as a device owner.

4. In the user settings window that opens, select the **Devices** tab.

5. Click **Add**.

6. From the device list, select the device that you want to assign to the user.

7. Click **OK**.

The selected device is added to the list of devices assigned to the user.

You can perform the same operation at **Asset management** → **Assets (Devices)**, by clicking the name of the device that you want to assign, and then clicking the **Manage device owner** link.

Configuring two-factor authentication

Starting with Kaspersky Next XDR Expert 2.0, two-factor authentication is enabled automatically. The global option to disable two-factor authentication is not supported. This scenario provides steps to configure two-step authentication at first sign-in and configure two-factor authentication settings on Administration Server.

Stages

Configuring two-factor authentication proceeds in stages:

1. **Installing an authenticator app on a device**

   You can install any application that supports the Time-Based One-Time Password algorithm (TOTP), such as:

   - Google Authenticator
   - Microsoft Authenticator
   - Bitrix24 OTP
   - Yandex ID
   - Avanpost Authenticator
   - Aladdin 2FA
   - Rutoken OTP

   To check if Kaspersky Next XDR Expert supports the authenticator app that you want to use, enable two-factor authentication for all users or for a particular user.

   One of the steps suggests that you specify the security code generated by the authenticator app. If it succeeds, then Kaspersky Next XDR Expert supports the selected authenticator.

We strongly do not recommend installing the authenticator app on the same device from which Administration Server is managed.

2. **Synchronizing the time on the device with the authenticator app with the time of the device on which Administration Server is installed**

   Ensure that the time on the device with the authenticator app and the time on the device with the Administration Server are synchronized to UTC. For better accuracy, we recommend using the same NTP servers throughout your infrastructure. Otherwise, failures may occur while configuring two-factor authentication.

3. **Configuring two-factor authentication for user accounts at sign-in to Web Console**

   Sign-in to Web Console and configure two-factor authentication.

   If two-factor authentication configuration is not available to you, contact a user who has the Modify object ACLs right of the **General features: User permissions** functional area and configured two-factor authentication to add your account to the two-factor authentication allow list.

4. **Configuring two-factor authentication settings on Administration Server**

   If you have configured two-factor authentication and the Modify object ACLs right of the **General features**: **User permissions** functional area, you can configure two-factor authentication settings on Administration Server as follows:

   1. Reviewing the list of users who can configure two-step authentication at sign-in:

      - For installing Kaspersky Next XDR Expert 2.0 from scratch: Add required users to the two-factor authentication allow list to enable them to configure two-factor authentication at his or her first sign-in.

      - For upgrading to Kaspersky Next XDR Expert 2.0: Review the two-factor authentication allow list and delete accounts that must not have access to Administration Server.

   2. Excluding user accounts for which you do not need to enable two-factor authentication (optional)

      Exclude user accounts from two-factor authentication to allow them to sign in to Administration Server even if they have not configured two-factor authentication. Excluding accounts from two-factor authentication may be necessary for integration accounts that cannot provide a security code during authentication. Integration accounts are used to run scripts through OpenAPI.

   3. Resetting or deleting a two-factor authentication secret key (optional)

      You can reset a two-factor authentication secret key when a user loses access to his or her two-factor authentication device, or needs to set up two-factor authentication on a new device. You can also reset a secret key for your own account.

      You can delete a secret key to completely block a user from signing in to Web Console and prevent the user from accessing Administration Server.

   4. Editing the name of a security code issuer (optional)

      If you have several Administration Servers with similar names, you may have to change the security code issuer names so there is better recognition of different Administration Servers.

Results

Upon completion of this scenario:

- Two-factor authentication is configured for your own account and for another user account who needs access to Administration Server.
- Integration accounts are excluded from two-factor authentication.

About two-factor authentication for an account

Kaspersky Next XDR Expert provides two-factor authentication for users of OSMP Console. Two-factor authentication is based on the RFC 6238 standard (TOTP: Time-Based One-Time Password algorithm).

Every time you log in to OSMP Console, you enter your user name, password, and an additional one-time security code. To receive a one-time security code, you must have an authenticator app on the mobile device or computer.

Any authentication software that supports the Time-Based One-Time Password algorithm (TOTP) can be used as an authenticator app, for example, Google Authenticator. In order to generate the one-time security code, you must synchronize the time set on the device with the authenticator app with the time set on the Administration Server device. For better accuracy, we recommend using the same NTP servers throughout your infrastructure.

To check if Kaspersky Next XDR Expert supports the authenticator app that you want to use, try configuring two-factor authentication with this authenticator app.

One of the steps suggests that you specify the one-time security code generated by the authenticator app. If it succeeds, then Kaspersky Next XDR Expert supports the selected authenticator.

Generating a secret code by using an authentication app

An authenticator app generates the security code as follows:

1. Administration Server generates a special secret key and QR code.
2. You pass the generated secret key or QR code to the authenticator app.
3. The authenticator app generates a single-use security code that you pass to the authentication window of Administration Server.

A security code has an identifier referred to as *issuer name*. The security code issuer name is used as an identifier of the Administration Server in the authenticator app. You can change the name of the security code issuer name. The security code issuer name has a default value that is the same as the name of the Administration Server. A new security code issuer name is assigned to Administration Server after the next signing-in. A security code is single-use and valid for up to 30 seconds (the exact time may vary).

We highly recommend that you save the secret key (or QR code) and keep it in a safe place. This will help you to restore access to OSMP Console in case you lose access to the device with the authenticator app.

Two-factor authentication has the following features:

- Starting with Kaspersky Next XDR Expert 2.0, two-factor authentication is enabled automatically and the option to configure two-factor authentication at sign-in is disabled by default. Configuring two-factor authentication at sign-in is available only to users who are included in the two-factor authentication allowlist.
- A user can configure two-factor authentication only for his or her own account.
- A user can exclude accounts from two-factor authentication. This can be necessary for integration accounts that cannot receive a one-time security code for authentication. Integration accounts are used to run scripts through OpenAPI and to run KUMA scripts.
- To regenerate a two-factor authentication secret key, a user must reset the secret key for his or her own account.
- To reset or delete a two-factor authentication secret key for account of another user, a user must have an account with configured two-factor authentication and with the Modify object ACLs right in the **General features**: **User permissions** functional area.
- To exclude accounts of other users from two-factor authentication or modify the two-factor authentication allowlist, a user must have an account with configured two-factor authentication and with the Modify object ACLs right in the **General features**: **User permissions** functional area.
- There should always be at least one user with configured two-factor authentication so that this user can manage the two-factor authentication allowlist and the exclusion list.
- Two-factor authentication is enabled automatically only for Kaspersky Next XDR Expert 2.0 or later. If other primary or secondary Administration Servers are running version 1.4 or earlier, their two-factor authentication settings will remain unchanged.
- A virtual Administration Server inherits two-factor authentication settings from the physical Administration Server so that if two-factor authentication is enabled on the physical Administration Server, it is also enabled on a virtual Administration Server. You cannot configure two-factor authentication settings on a virtual Administration Server.

To ensure comprehensive protection by using two-factor authentication, it is necessary to secure not only Administration Server but also the device on which it is installed. To do this, consider the Hardening guide recommendations.

Configuring a two-factor authentication secret key at OSMP Console sign-in

The first time you sign in to Kaspersky Next XDR Expert 2.0 or later, the window for configuring two-factor authentication opens.

To configure two-factor authentication for your account:

1. Install any application that supports the Time-Based One-Time Password algorithm (TOTP) on the mobile device or computer, such as:
   - Google Authenticator
   - Microsoft Authenticator

- Bitrix24 OTP
- Yandex ID
- Avanpost Authenticator
- Aladdin 2FA
- Rutoken OTP

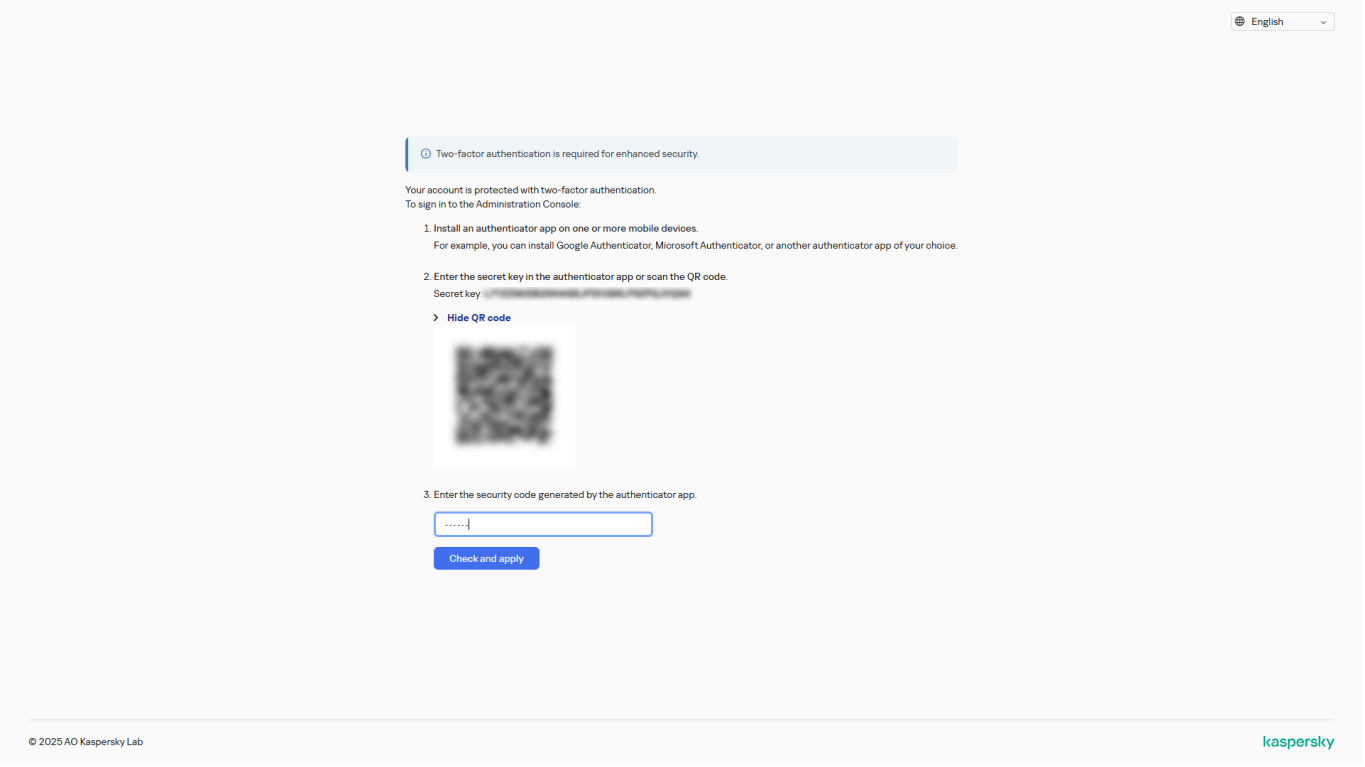
> We strongly do not recommend installing the authenticator app on the same device from which Administration Server is managed.

2. Ensure that the time on the device with the authenticator app and the time on the device with the Administration Server are synchronized to UTC. For better accuracy, we recommend using the same NTP servers throughout your infrastructure.

3. Generate a one-time security code by using the authenticator app on the mobile device or computer. To do this, perform one of the following actions:

   - Enter the two-factor authentication secret key in the authenticator app manually.
   - Click **View QR code** and scan the QR code by using the authenticator app.

   A one-time security code of 6 digits will display on the mobile device.



Configuring a two-factor authentication secret key at OSMP Console sign-in

4. In the **Two-factor authentication** window, specify the one-time security code generated by the authenticator app, and then click the **Check and apply** button.

   Rutoken OTP configuration

If you use Rutoken OTP, you need an Android mobile device with the Rutoken OTP application installed. The Android mobile device must the following requirements:

- Operating system is Android 7 or newer.
- NFC module is available and enabled.

In the Rutoken OTP application, you have to configure the following parameters:

- Scan the QR code displayed in the two-step configuration wizard.
- Set the time interval to 30 seconds.
- Select SHA1 as the algorithm.

To configure Rutoken OTP:

1. Put the Rutoken OTP device next to the mobile device with the NFC module.
2. In the Rutoken OTP application, press **Connect token**.
3. In the Rutoken OTP application, press **Configure settings**, and then press **Save secret key.**

   If the token is successfully connected to the mobile device, the Rutoken OTP application shows the following status messages:

   - **Token is connected.**
   - **Settings are configured.**
   - **Secret key is saved.**
4. On the Rutoken OTP device, press the button.

   A security code is displayed on the device.
5. In OSMP Console, in the two-step verification window, enter the security code.

Two-factor authentication is configured for your account. You are able to access the Administration Server in accordance with your rights.

If two-factor authentication configuration is not available to you, contact a user who has the Modify object ACLs right in the **General features**: **User permissions** functional area and configured two-factor authentication to add your account to the two-factor authentication allowlist.

Allowing users to configure two-factor authentication

The two-factor authentication allowlist is a list of users who are permitted to initially configure two-factor authentication when signing in to Web Console. This list contains user accounts without configured two-factor authentication. By default, new user accounts are not included in the two-factor authentication allowlist and must be added to the list by an administrator. If a new account is an integration account used to run OpenAPI scripts, it must be excluded from two-factor authentication.

A user account will be automatically deleted from the two-factor authentication allowlist when it has configured two-factor authentication or is excluded from two-factor authentication.

Manual modifying the two-factor authentication allowlist is available for accounts that have configured two-factor authentication and that have the Modify object ACLs right in the **General features**: **User permissions** functional area.

To add a user account to the allow list manually:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **Two-factor authentication** tab of the properties window, in the **Users allowed to generate secret keys** table, click the **Add** button.

3. In the window that opens:

   a. Select the user account that you want to add.

      Only users who are not excluded from two-factor authentication and do not have a two-factor authentication secret key can be added to the allow list.

   b. If the account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and one-time security code for two-factor authentication.

   c. Save the changes.

   The selected user account is added to the two-factor authentication allowlist. At the next sign-in, the user will be able to generate a secret key.

A user account can be deleted from the two-factor authentication allowlist until two-factor authentication is configured for that account.

To delete user accounts from the two-factor authentication allowlist manually:

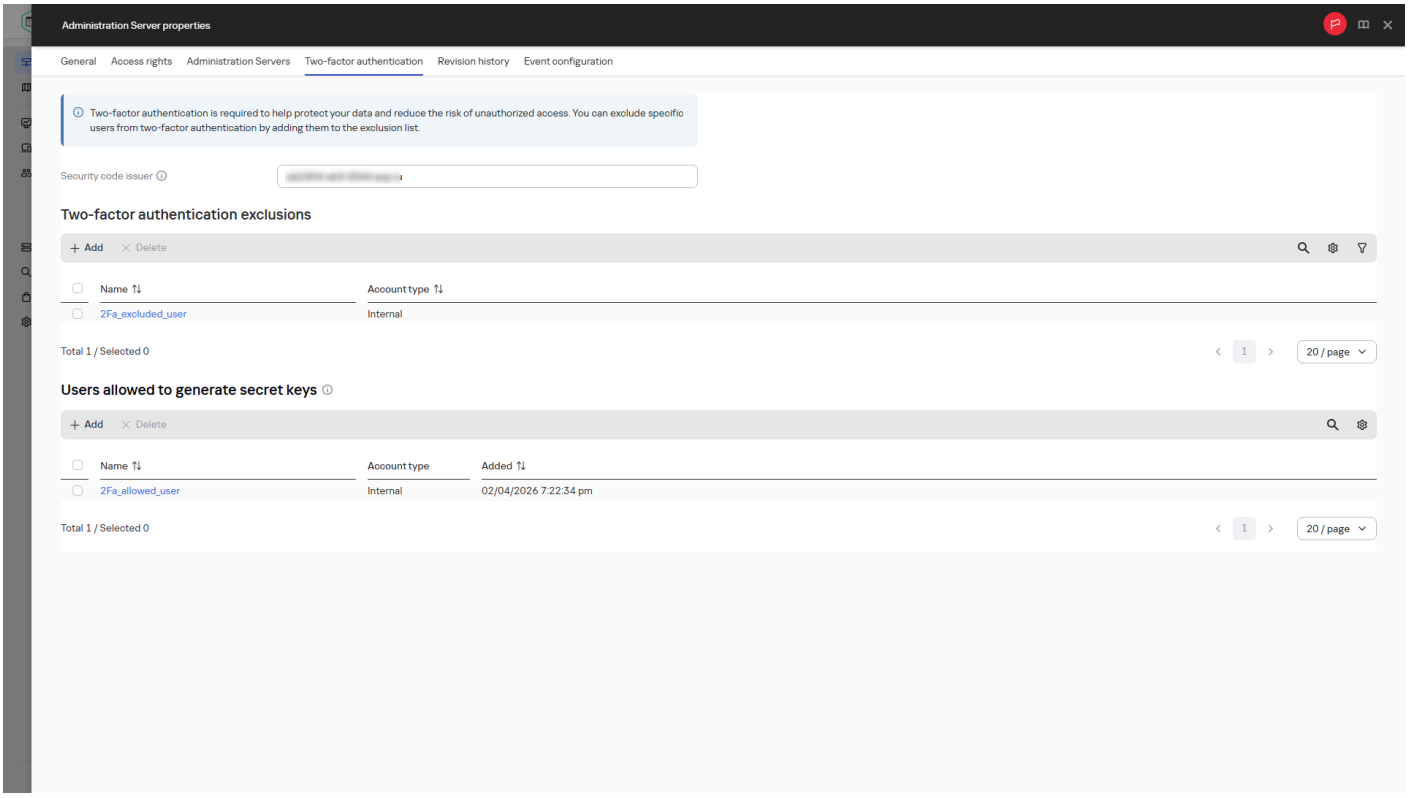
1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **Two-factor authentication** tab of the properties window, in the **Users allowed to generate secret keys** table, select the user accounts that you want to delete, and then click **Delete**.

3. If the account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and one-time security code for two-factor authentication.

The selected user accounts are deleted from the allow list.



Configuring the two-factor authentication allowlist

If, during an active session on Administration Server, your two-factor authentication secret key is deleted by another user or you reset the key yourself but do not create a new one, the ability to modify the allow list is blocked.

User accounts can also be automatically added and deleted from the two-factor authentication allowlist.

When resetting a two-factor authentication secret key for other accounts, their accounts are automatically added to the two-factor authentication allowlist so that they can generate a new secret key.

When upgrading Kaspersky Next XDR Expert from version 1.4 to version 2.0 or later, user accounts will be automatically added to the two-factor authentication allow list if all the following conditions are met:

- User has the Read permission in the functional area *General features: Basic functionality* assigned directly to the user (not through a security group). The scope of the user role with this permission must include all managed and unassigned devices, and must be set at the Administration Server level.

- User is not excluded from two-factor authentication.

- User does not have a two-factor authentication secret key.

- Two-factor authentication was not enabled for all users on Administration Server of the previous version.

Failure to meet even one condition will result in the user not being added to the two-factor authentication allowlist.

> After an upgrade or restore, we recommend ensuring that the allow list contains only the user accounts that must have access to Administration Server. Review the allow list and remove any other accounts from it. Add integration accounts used to run scripts through OpenAPI to the exclusion list to prevent script failures.

Accounts of the users will lose access to Administration Server after an upgrade or restore if all the following conditions are met:

- User has the Read permission in the functional area *General features: Basic functionality* assigned through a security group (not directly to the user) or does not have this permission at all.
- User is not excluded from two-factor authentication.
- User does not have a two-factor authentication secret key.
- Two-factor authentication was enabled for all users on Administration Server of the previous version.

These accounts can be manually added to the two-factor authentication allowlist, if needed.

Resetting and deleting a two-factor authentication secret key

You can reset a secret key when a user loses access to his or her two-factor authentication device or needs to set up two-factor authentication on a new device. Resetting the secret key allows the user to sign in to Web Console again and generate a new secret key.

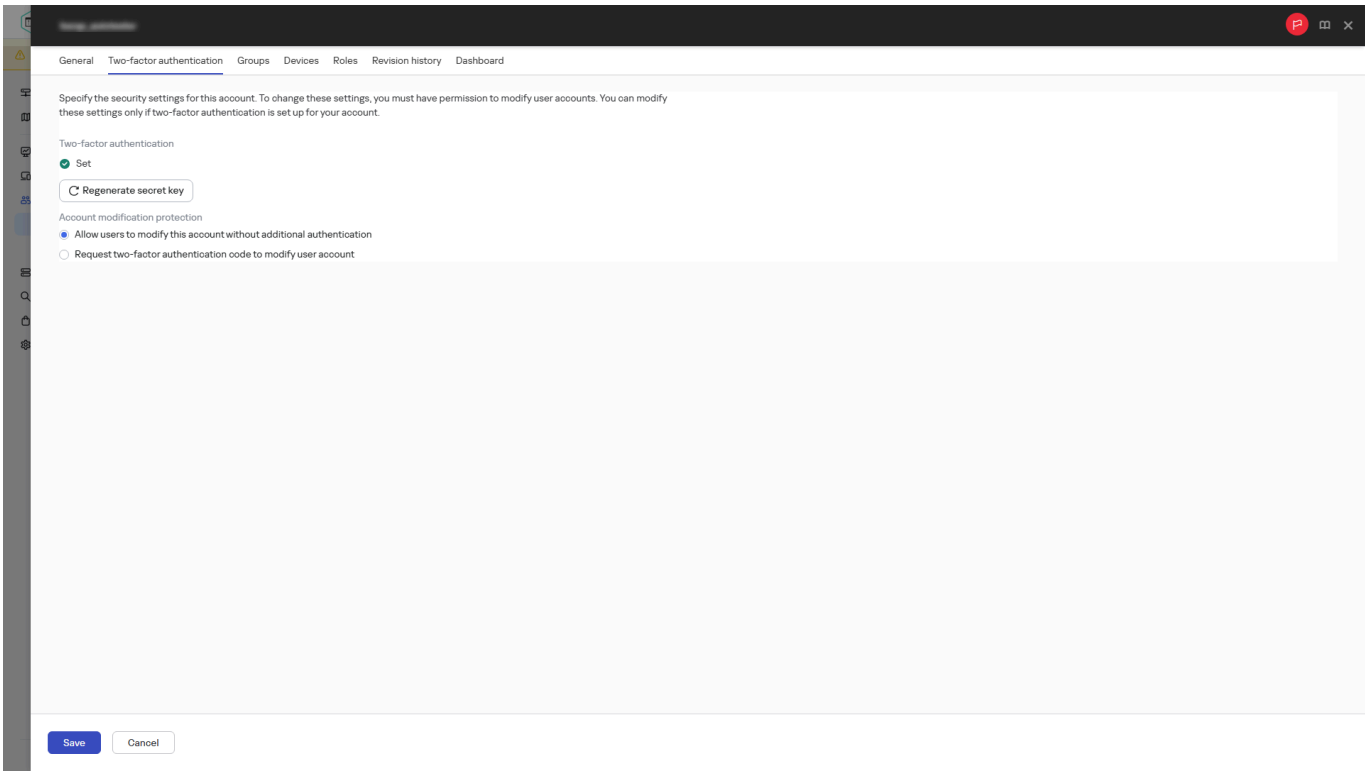
> You can reset or delete a two-factor authentication secret key for accounts of other users only if you have configured two-factor authentication and you have the Modify object ACLs right in the **General features**: **User permissions** functional area. You can also regenerate your own secret key.

To reset a secret key for an account:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.
2. Click the name of the user account for which you want to reset a secret key.

3. If you want to reset a secret key for your account:

   a. On the **Two-factor authentication** tab, click the **Regenerate secret key** button.

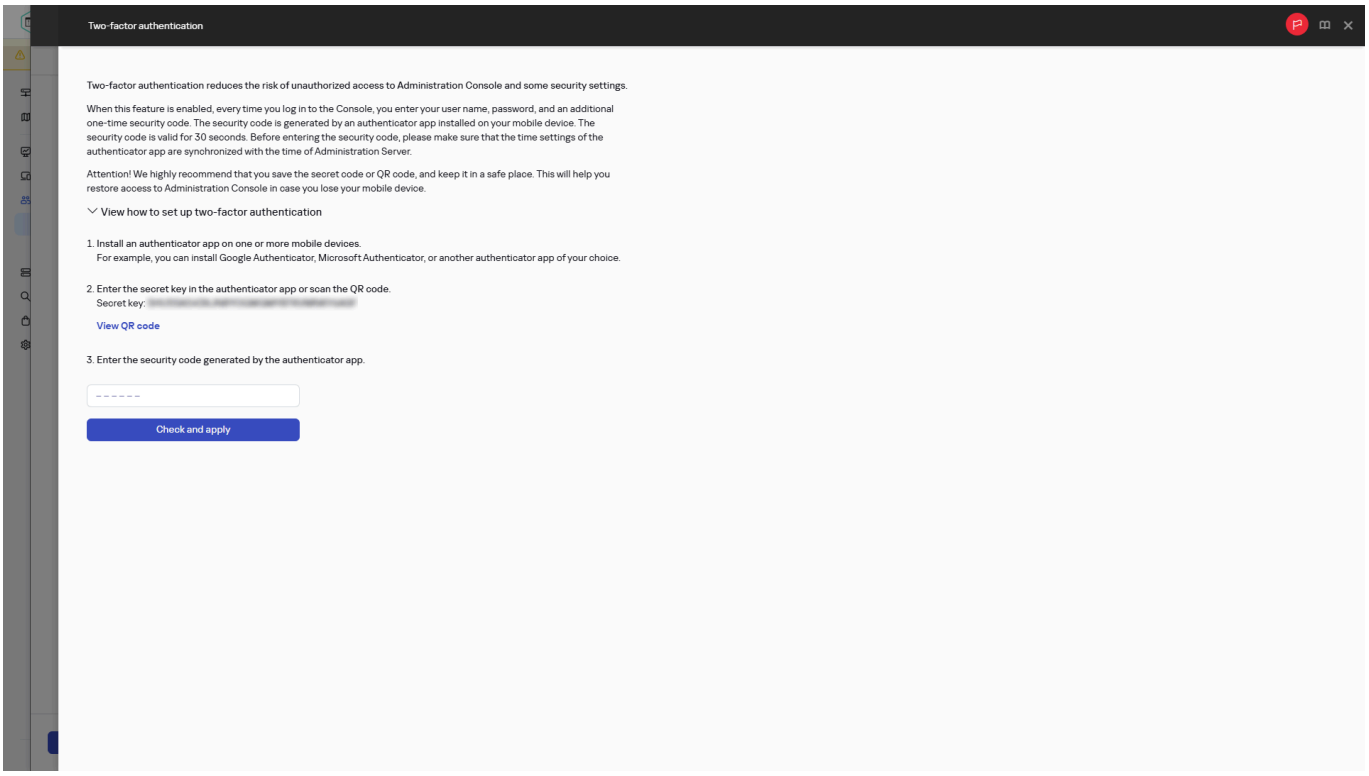


Regenerating the two-factor authentication secret key for the current user account

   a. If your account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and the two-factor authentication code.

   b. In the **Two-factor authentication** window, click **View how to set up two-factor authentication**, and then click **View QR code**.

   c. Use the authenticator app on the mobile device to scan the QR code and receive a one-time security code.



Configuring a new two-factor authentication secret key for the current user account

a. In the **Two-factor authentication** window, specify the one-time security code generated by the authenticator app, click the **Check and apply** button, and then save changes.

Rutoken OTP configuration

If you use Rutoken OTP, you need an Android mobile device with the Rutoken OTP application installed. The Android mobile device must the following requirements:

- Operating system is Android 7 or newer.
- NFC module is available and enabled.

In the Rutoken OTP application, you have to configure the following parameters:

- Scan the QR code displayed in the two-step configuration wizard.
- Set the time interval to 30 seconds.
- Select SHA1 as the algorithm.

To configure Rutoken OTP:

1. Put the Rutoken OTP device next to the mobile device with the NFC module.
2. In the Rutoken OTP application, press **Connect token**.
3. In the Rutoken OTP application, press **Configure settings**, and then press **Save secret key.**

If the token is successfully connected to the mobile device, the Rutoken OTP application shows the following status messages:

- **Token is connected.**
- **Settings are configured.**
- **Secret key is saved.**

4. On the Rutoken OTP device, press the button.

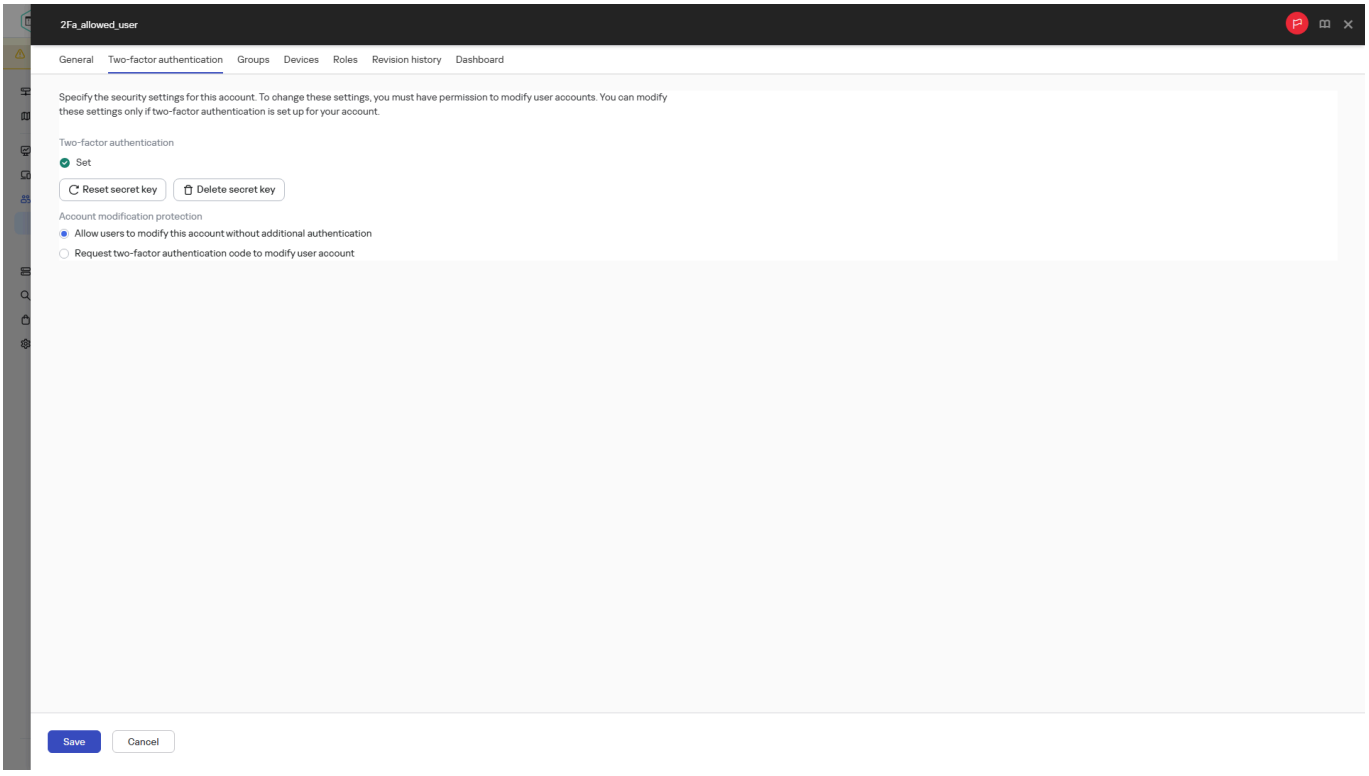
   A security code is displayed on the device.

5. In OSMP Console, in the two-step verification window, enter the security code.

The secret key for your account is changed.

4. If you want to reset a secret key for another account:

   a. On the **Two-factor authentication** tab, click the **Reset secret key** button.



Resetting the two-factor authentication secret key for another user account

   a. If the account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and the one-time security code for two-factor authentication.

   b. Confirm the reset of the secret key, and then save changes.

When you reset a secret key for another account, the existing secret key is deleted and the user account is added to the two-factor authentication allowlist in the Administration Server properties. At the next sign-in, the user will be able to generate a new secret key.

The secret key of a user account is reset.

You can delete a secret key to completely block a user from signing in to Web Console and prevent the user from accessing Administration Server.

To delete a secret key:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.
2. Click the name of the internal user account for which you want to delete a secret key.

   You cannot delete a secret key for your own account.

3. On the **Two-factor authentication** tab, click the **Delete secret key** button.
4. If the account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and the one-time security code for two-factor authentication.
5. Confirm the deletion of the secret key, and then save changes.

The secret key of a user account is deleted.

You can restore access by adding the user to the two-factor authentication allowlist in the Administration Server properties. In this case, at the next sign-in, the user will be able to generate a new secret key.

Excluding accounts from two-factor authentication

You can exclude user accounts from two-factor authentication to allow them to sign in to Administration Server even if they have not configured two-factor authentication. Excluding accounts from two-factor authentication may be necessary for integration accounts that cannot provide a one-time security code during authentication. Integration accounts are used to run scripts through OpenAPI and to run KUMA scripts.

You can modify the exclusion list if two-factor authentication is configured for your account and you have the Modify object ACLs right in the **General features**: **User permissions** functional area.

To exclude a user account from two-factor authentication:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **Two-factor authentication** tab of the properties window, in the **Two-factor authentication exclusions** table, click the **Add** button.

3. In the window that opens:

   a. Select the user account that you want to exclude.

   b. If the account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and one-time security code for two-factor authentication.

   c. Save the changes.

   The selected user accounts are excluded from two-factor authentication, and their secret keys are deleted if they exist. The list of excluded accounts is not changed when you upgrade Kaspersky Next XDR Expert to version 2.0 or later. If the accounts from the exclusion list had secrets in the previous version of the Kaspersky Next XDR Expert, these secret keys will be deleted in 2.0.

You can delete user accounts from the exclusion list to completely block users from signing in to Web Console and to prevent users from accessing Administration Server.

To delete user accounts from the exclusion list:

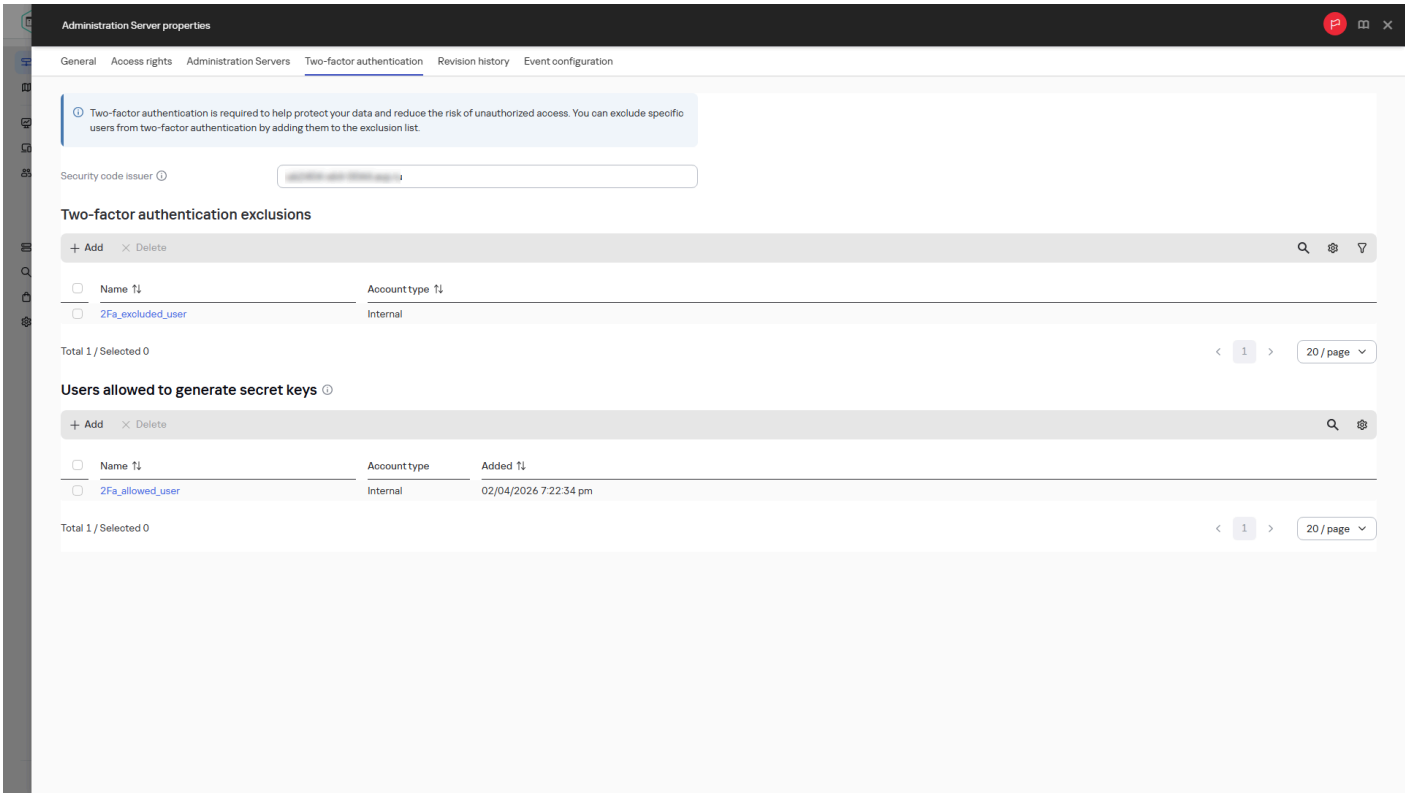
1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **Two-factor authentication** tab of the properties window, in the **Two-factor authentication exclusions** table, select the user accounts that you want to delete, and then click **Delete**.

3. If the account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of your own account and one-time security code for two-factor authentication.

   The selected user accounts are deleted from the exclusion list.



Configuring the two-factor authentication exclusion list

Editing the name of a security code issuer

You can have several identifiers (called issuers) for different Administration Servers. You can change the name of a security code issuer in case, for example, the Administration Server already uses a similar name of the security code issuer for another Administration Server. By default, the name of a security code issuer is the same as the name of the Administration Server.

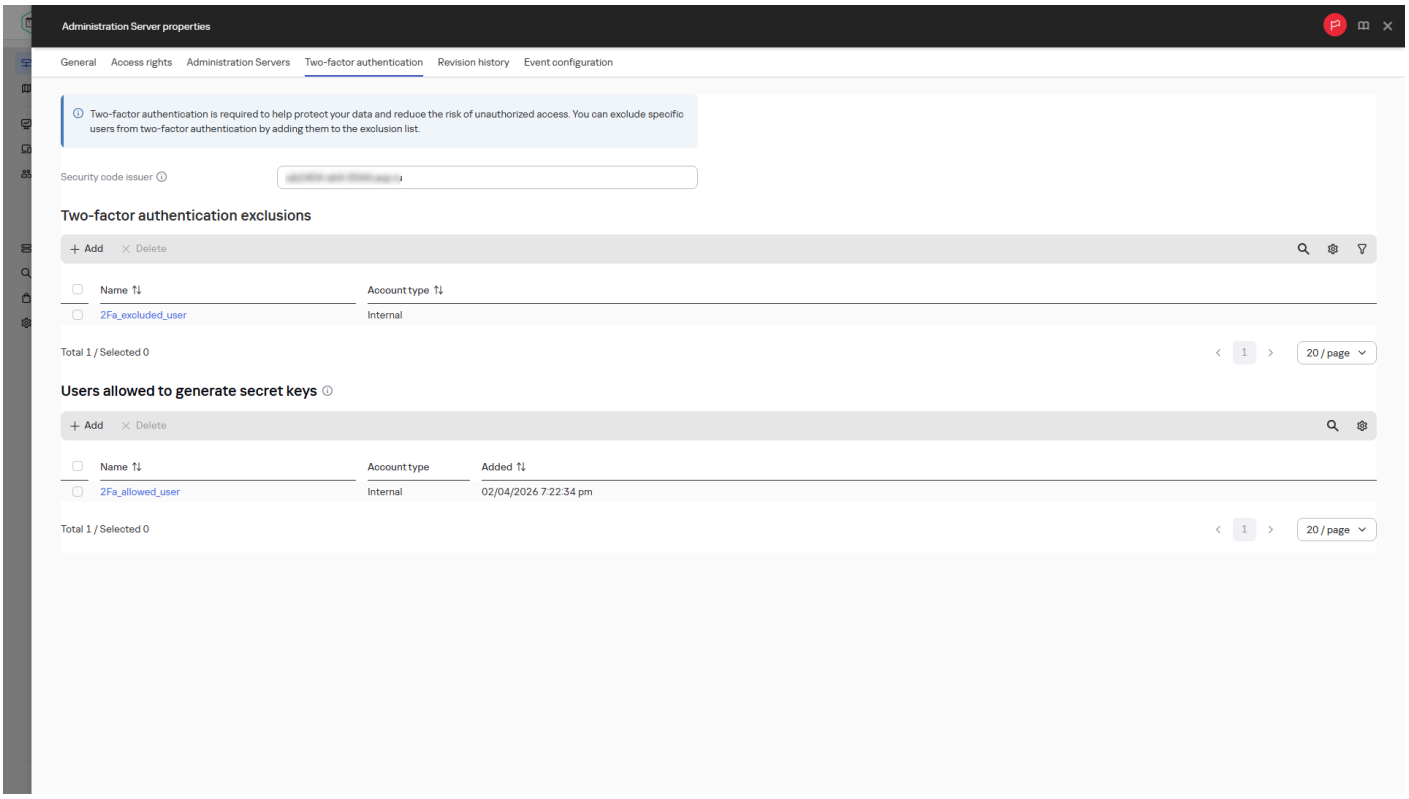
To specify a new name of a security code issuer:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **Two-factor authentication** tab of the properties window, enter a new name in the **Security code issuer** field, and then save changes.

A new security code issuer name is assigned to Administration Server after the next sign-in.



Specifying the name of a security code issuer

Changing the number of allowed password entry attempts

The Open Single Management Platform user can enter an invalid password a limited number of times. After the limit is reached, the user account is blocked for one hour.

By default, the maximum number of allowed attempts to enter a password is 5. This number is fixed and cannot be changed.

Deleting a user or a security group

You can delete only internal users or internal security groups.

To delete a user or a security group:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** or the **Groups** tab.
2. Select the check box next to the user or the security group that you want to delete.

   There should always be at least one user with configured two-factor authentication so that this user can manage the two-factor authentication allowlist and the exclusion list.

3. Click **Delete**.
4. In the window that opens, click **OK**.

The user or the security group is deleted.

Changing the password for a user account

You may need to change the password for a user account if the current password is about to expire or you want to change it to a more secure one.

The password must comply with the following rules:

- The password must be 8 to 256 characters long.
- The password must contain characters from at least three of the groups listed below:
  - Uppercase letters (A-Z)
  - Lowercase letters (a-z)
  - Numbers (0-9)
  - Special characters (@ # $ % ^ & * - _ ! + = [ ] { } | : ' , . ? / \ ` ~ " ( ) ;)
- The password must not contain any whitespaces, Unicode characters, or the combination of "." and "@", when "." is placed before "@".

Changing the password for an internal user account

To edit the password for an internal user account:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.
2. Click the name of the user account that you want to edit.
3. In the user settings window that opens, on the **General** tab, click the **Change password** button.
4. Set a new password for the user connection to Open Single Management Platform.

   To see the entered password, click 👁.
5. If the user account is protected against unauthorized modification, you must confirm that you have the permissions to change this account. In the **Account protection** window, specify the credentials of the account that has the Modify object ACLs right in the General features: User permissions functional area.
6. Enter the one-time security code generated by the authenticator app on the mobile device.

Creating a user role

To create a user role:

1. In the main menu, go to **Users & roles** → **Roles**.
2. Click **Add**.
3. In the **New role name** window that opens, enter the name of the new role.
4. Click **OK** to apply the changes.
5. In the role properties window that opens, change the settings of the role:
   - On the **General** tab, edit the role name.

     You cannot edit the name of a predefined role.
   - On the **Settings** tab, edit the role scope and policies and profiles associated with the role.
   - On the **Access rights** tab, edit the rights for access to Kaspersky applications.
6. Click **Save** to save the changes.

The new role appears in the list of user roles.

Editing a user role

To edit a user role:

1. In the main menu, go to **Users & roles** → **Roles**.
2. Click the name of the role that you want to edit.
3. In the role properties window that opens, change the settings of the role:
   - On the **General** tab, edit the role name.
     You cannot edit the name of a predefined role.
   - On the **Settings** tab, edit the role scope and policies and profiles associated with the role.
   - On the **Access rights** tab, edit the rights for access to Kaspersky applications.
4. Click **Save** to save the changes.

The updated role appears in the list of user roles.

Editing the scope of a user role

A *user role scope* is a combination of users and administration groups. Settings associated with a user role apply only to devices that belong to users who have this role, and only if these devices belong to groups associated with this role, including child groups.

To add users, security groups, and administration groups to the scope of a user role, you can use either of the following methods:

Method 1:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** or the **Groups** tab.
2. Select check boxes next to the users or security groups that you want to add to the user role scope.
3. Click the **Assign role** button.
   The Role assignment wizard starts. Proceed through the wizard by using the **Next** button.
4. On the **Select role** step, select the user role that you want to assign.
5. On the **Define scope** step, select the administration group that you want to add to the user role scope.
6. Click the **Assign role** button to close the window.

The selected users or security groups and the selected administration group are added to the scope of the user role.

Method 2:

1. In the main menu, go to **Users & roles** → **Roles**.
2. Click the name of the role for which you want to define the scope.
3. In the role properties window that opens, select the **Settings** tab.
4. In the **Role scope** section, click **Add**.

   The Role assignment wizard starts. Proceed through the wizard by using the **Next** button.
5. On the **Define scope** step, select the administration group that you want to add to the user role scope.
6. On the **Select users** step, select users and security groups that you want to add to the user role scope.
7. Click the **Assign role** button to close the window.
8. Click the **Close** button (×) to close the role properties window.

The selected users or security groups and the selected administration group are added to the scope of the user role.

Method 3:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.
2. On the **Access rights** tab, select the check box next to the name of the user or the security group that you want to add to the user role scope, and then click the **Roles** button.

   You cannot select multiple users or security groups at the same time. If you select more than one item, the **Roles** button will be disabled.
3. In the **Roles** window, select the user role that you want to assign, and then apply and save changes.

   The selected users or security groups are added to the scope of the user role.

Deleting a user role

To delete a user role:

1. In the main menu, go to **Users & roles** → **Roles**.
2. Select the check box next to the name of the role that you want to delete.
3. Click **Delete**.
4. In the window that opens, click **OK**.

The user role is deleted.

Associating policy profiles with roles

You can associate user roles with policy profiles. In this case, the activation rule for this policy profile is based on the role: the policy profile becomes active for a user that has the specified role.

For example, the policy bars any GPS navigation software on all devices in an administration group. GPS navigation software is necessary only on a single device in the Users administration group—the device owned by a courier. In this case, you can assign a "Courier" role to its owner, and then create a policy profile allowing GPS navigation software to run only on the devices whose owners are assigned the "Courier" role. All the other policy settings are preserved. Only the user with the role "Courier" will be allowed to run GPS navigation software. Later, if another worker is assigned the "Courier" role, the new worker also can run navigation software on your organization's device. Running GPS navigation software will still be prohibited on other devices in the same administration group.

To associate a role with a policy profile:

1. In the main menu, go to **Users & roles** → **Roles**.
2. Click the name of the role that you want to associate with a policy profile.

   The role properties window opens with the **General** tab selected.
3. Select the **Settings** tab, and scroll down to the **Policies** section.
4. Click **Edit**.
5. To associate the role with:
   - **An existing policy profile**—Click the chevron icon (>) next to the required policy name, and then select the check box next to the profile with which you want to associate the role.
   - **A new policy profile**:

     a. Select the check box next to the policy for which you want to create a profile.

     b. Click **New policy profile**.

     c. Specify a name for the new profile and configure the profile settings.

     d. Click the **Save** button.

     e. Select the check box next to the new profile.
6. Click **Assign to role**.

The profile is associated with the role and appears in the role properties. The profile applies automatically to any device whose owner is assigned the role.

Configuring Kerberos domain authentication

Open Single Management Platform enables you to use domain authentication over the Kerberos protocol. Domain authentication allows you to enable secure authentication in OSMP Console without having to re-enter the password on the corporate network (single sign-on).

You must first configure your domain controller to use Kerberos and generate a keytab file. For more information, refer to the documentation for your domain controller.

SCRIL (Smart card is required for interactive logon) is supported for Microsoft Active Directory.

Generating a keytab file for FreeIPA and AldPro

To generate a keytab file:

1. Create a user account to be used by IAM to access the domain controller.

2. If the Administration Server is outside the FreeIPA/AldPro domain, add the DNS record entry and the host entry:

```
ipa dnsrecord-add <zoneName> <recordName> --a-rec <Administration Server IP address>
ipa host-add --password=<password> <Administration Server host address>
```

3. Specify the PAC type when creating the HTTP service for the Administration Server:

```
ipa service-add HTTP/<Administration Server host FQDN>@<DOMAIN WRITTEN IN CAPITAL LETTERS> --ok-to-auth-as-
delegate=true --pac-type PAD
```

4. Specify the user account for authentication:

```
ipa service-allow-retrieve-keytab HTTP/<Administration Server host address>@<DOMAIN WRITTEN IN CAPITAL LETTERS>
--user=<user>
```

5. Export the keytab file:

```
ipa-getkeytab -s <domain controller address> -p HTTP/<Administration Server host address>@<DOMAIN WRITTEN IN
CAPITAL LETTERS> -k /tmp/h.keytab
```

Generating a keytab file for Microsoft Active Directory

To generate a keytab file:

1. Create a user account to be used by IAM to communicate with the domain controller.

2. Enable the following options for the user account:

   - **User cannot change password**

   - **Kerberos AES 256-bit encryption**

   You can enable the **Password never expires** option to avoid reissuing the keytab file each time the password expires.

3. Add a Service Principal Name (SPN) to the user account:

```
setspn -S HTTP/<Administration Server host FQDN> <DOMAIN WRITTEN IN CAPITAL LETTERS>\<user>
```

4. Export the keytab file:

```
ktpass -out C:\<folder>\<keytab file name> -princ HTTP/<Administration Server host FQDN>@<DOMAIN WRITTEN IN
CAPITAL LETTERS> -mapuser <DOMAIN WRITTEN IN CAPITAL LETTERS>\<user> -crypto AES256-SHA1 -ptype
KRB5_NT_PRINCIPAL -pass <user password> -mapop set
```

Generating a keytab file for Samba

To generate a keytab file:

1. Create a user account to be used by IAM to communicate with the domain controller:

```
samba-tool user add <user>
```

2. Add the SPN to the user account:

```
samba-tool spn add HTTP/<Administration Server host FQDN> <user>
```

3. Export the keytab file:

```
samba-tool domain exportkeytab /tmp/file.keytab --principal=HTTP/<Administration Server host FQDN>
```

Enabling Kerberos domain authentication

To enable Kerberos domain authentication:

1. Perform domain controller polling.
2. In the main menu, go to **Settings** → **Single sign-on**.
3. Enable the **Kerberos authentication** option.
4. Upload the keytab file.
5. Assign a role to domain users that gives access to OSMP Console.

   Kerberos domain authentication is enabled for domain users.

Updating Kaspersky databases and applications

This section describes steps you must take to regularly update the following:

- Kaspersky databases and software modules
- Installed Kaspersky applications, including Open Single Management Platform components and security applications

Updates functionality (including providing anti-virus signature updates and codebase updates), as well as KSN functionality may not be available in the software in the U.S.

Scenario: Regular updating Kaspersky databases and applications

This section provides a scenario for regular updating of Kaspersky databases, software modules, and applications. After you complete the Configuring network protection scenario, you must maintain the reliability of the protection system to make sure that the Administration Servers and managed devices are kept protected against various threats, including viruses, network attacks, and phishing attacks.

Network protection is kept up-to-date by regular updates of the following:

- Kaspersky databases and software modules
- Installed Kaspersky applications, including Open Single Management Platform components and security applications

When you complete this scenario, you can be sure of the following:

- Your network is protected by the most recent Kaspersky software, including Open Single Management Platform components and security applications.
- The anti-virus databases and other Kaspersky databases critical for the network safety are always up-to-date.

Prerequisites

The managed devices must have a connection to the Administration Server. If they do not have a connection, consider updating Kaspersky databases and software modules manually or directly from the Kaspersky update servers.

Administration Server must have a connection to the internet.

Before you start, make sure that you have done the following:

1. Deployed the Kaspersky security applications to the managed devices according to the scenario of deploying Kaspersky applications through OSMP Console.
2. Created and configured all required policies, policy profiles, and tasks according to the scenario of configuring network protection.
3. Assigned an appropriate amount of distribution points in accordance with the number of managed devices and the network topology.

Updating Kaspersky databases and applications proceeds in stages:

1. **Choosing an update scheme**

   There are several schemes that you can use to install updates to Open Single Management Platform components and security applications. Choose the scheme or several schemes that meet the requirements of your network best.

2. **Creating the task for downloading updates to the repository of the Administration Server**

   Create the *Download updates to the Administration Server repository* task manually.

   This task is required to download updates from Kaspersky update servers to the repository of the Administration Server, as well as to update Kaspersky databases and software modules for Open Single Management Platform. After the updates are downloaded, they can be propagated to the managed devices.

   If your network has assigned distribution points, the updates are automatically downloaded from the Administration Server repository to the repositories of the distribution points. In this case the managed devices included in the scope of a distribution point download the updates from the repository of the distribution point instead of the Administration Server repository.

   How-to instructions: Creating the task for downloading updates to the repository of the Administration Server

3. **Creating the task for downloading updates to the repositories of distribution points (optional)**

   By default, the updates are downloaded to the distribution points from the Administration server. You can configure Open Single Management Platform to download the updates to the distribution points directly from Kaspersky update servers. Download to the repositories of distribution points is preferable if the traffic between the Administration Server and the distribution points is more expensive than the traffic between the distribution points and Kaspersky update servers, or if your Administration Server does not have internet access.

   When your network has assigned distribution points and the *Download updates to the repositories of distribution points* task is created, the distribution points download updates from Kaspersky update servers, and not from the Administration Server repository.

   How-to instructions: Creating the task for downloading updates to the repositories of distribution points

4. **Configuring distribution points**

   When your network has assigned distribution points, make sure that the **Deploy updates** option is enabled in the properties of all required distribution points. When this option is disabled for a distribution point, the devices included in the scope of the distribution point download updates from the repository of the Administration Server.

5. **Optimizing the update process by using the diff files (optional)**

   You can optimize traffic between the Administration Server and the managed devices by using diff files. When this feature is enabled, the Administration Server or a distribution point downloads diff files instead of entire files of Kaspersky databases or software modules. A diff file describes the differences between two versions of a file of a database or software module. Therefore, a diff file occupies less space than an entire file. This results in decrease in the traffic between the Administration Server or distribution points and the managed devices. To use this feature, enable the **Download diff files** option in the properties of the *Download updates to the Administration Server repository* task and/or the *Download updates to the repositories of distribution points* task.

   How-to instructions: Using diff files for updating Kaspersky databases and software modules

6. **Configuring automatic installation of updates for the security applications**

   Create the *Update* tasks for the managed applications to provide timely updates to the software modules and Kaspersky databases, including anti-virus databases. To ensure timely updates, we recommend that you select the **When new updates are downloaded to the repository** option when configuring the task schedule.

   If an update requires reviewing and accepting the terms of the End User License Agreement, then you first need to accept the terms. After that the update can be propagated to the managed devices.

7. **Approving and declining updates of managed Kaspersky applications**

By default, the downloaded software updates have the *Undefined* status. You can change the status to *Approved* or *Declined*. The approved updates are always installed. If an update of a managed Kaspersky application requires reviewing and accepting the terms of the End User License Agreement, then you first need to accept the terms. After that the update can be propagated to the managed devices. The updates for which you set *Declined* status will not be installed on devices. If a declined update for a managed application was previously installed, Open Single Management Platform will try to uninstall the update from all devices.

Approving and declining updates is available only for Network Agent and managed Kaspersky applications installed on Windows-based client devices. Seamless updating of Administration Server, OSMP Console, and management web plug-ins is not supported.

How-to instructions: Approving and declining software updates

Sometimes after Network Agent update you may be prompted to restart managed devices. If the notification about the required restart is displayed, click the **View devices** link in the notification, select the check boxes next to the devices to be restarted, and then click the **Restart** button to finish the Network Agent update.

Results

Upon completion of the scenario, Open Single Management Platform is configured to update Kaspersky databases after the updates are downloaded to the repository of the Administration Server. You can then proceed to monitoring the network status.

About updating Kaspersky databases, software modules, and applications

To be sure that the protection of your Administration Servers and managed devices is up-to-date, you must provide timely updates of the following:

- Kaspersky databases and software modules

  Before downloading Kaspersky databases and software modules, Open Single Management Platform checks if Kaspersky servers are accessible. If access to the servers using system DNS is not possible, the application uses public DNS servers. This is necessary to make sure anti-virus databases are updated and the level of security is maintained for the managed devices.

- Installed Kaspersky applications, including Open Single Management Platform components and security applications

  Open Single Management Platform allows you to update Network Agent and Kaspersky applications installed on Windows-based client devices automatically. Seamless updating of Administration Server, OSMP Console, and management web plug-ins is not supported. To update these components, you have to download the latest versions from the Kaspersky website, and then install them manually.

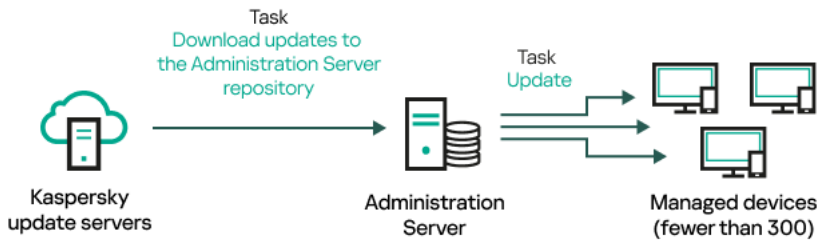
Depending on the configuration of your network, you can use the following schemes of downloading and distributing the required updates to the managed devices:

- By using a single task: *Download updates to the Administration Server repository*
- By using two tasks:
  - The *Download updates to the Administration Server repository* task
  - The *Download updates to the repositories of distribution points* task
- Manually through a shared folder or an FTP server

- Directly from Kaspersky update servers to Kaspersky Endpoint Security on the managed devices
- Through a network folder if Administration Server has no internet connection

Using the Download updates to the Administration Server repository task

In this scheme, Open Single Management Platform downloads updates through the *Download updates to the Administration Server repository* task. In small networks that contain less than 300 managed devices in a single network segment or less than 10 managed devices in each network segment, the updates are distributed to the managed devices directly from the Administration Server repository (see figure below).



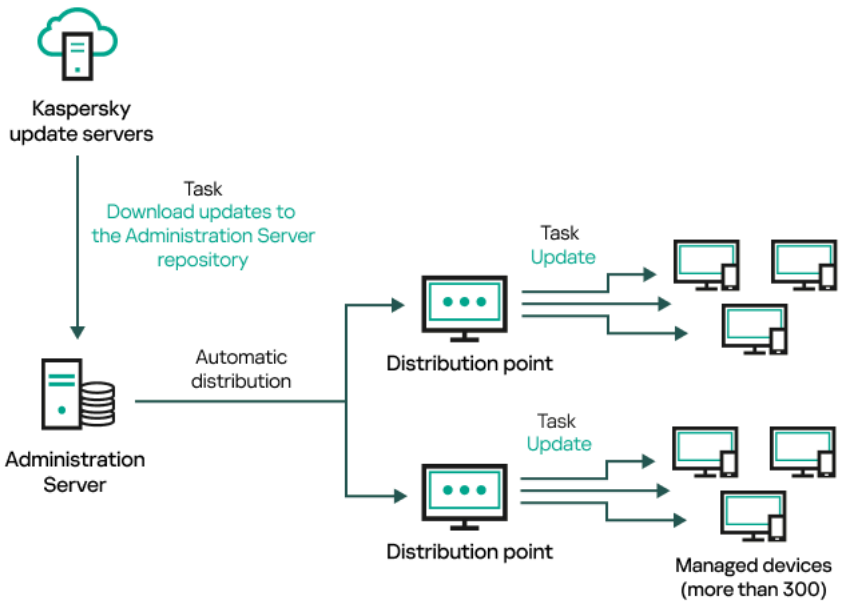
Updating by using the *Download updates to the Administration Server repository* task without distribution points

As a source of updates, you can use not only Kaspersky update servers, but also a network folder.

> By default, the Administration Server communicates with Kaspersky update servers and downloads updates by using the HTTPS protocol. You can configure the Administration Server to use the HTTP protocol instead of HTTPS.

If your network contains 300 managed devices or more in a single network segment or if your network consists of several network segments with more than 9 managed devices in each network segment, we recommend that you use distribution points to propagate the updates to the managed devices (see figure below). Distribution points reduce the load on the Administration Server and optimize traffic between the Administration Server and the managed devices. You can calculate the number and configuration of distribution points required for your network.

In this scheme, the updates are automatically downloaded from the Administration Server repository to the repositories of the distribution points. The managed devices included in the scope of a distribution point download the updates from the repository of the distribution point instead of the Administration Server repository.



Updating by using the *Download updates to the Administration Server repository* task with distribution points

When the *Download updates to the Administration Server repository* task is complete, the updates for Kaspersky databases and software modules for Kaspersky Endpoint Security are downloaded to the Administration Server repository. These updates are installed through the *Update* task for Kaspersky Endpoint Security.

> The *Download updates to the repository of the Administration Server* task is not available on virtual Administration Servers. The repository of the virtual Administration Server displays updates downloaded to the primary Administration Server.

You can configure the updates to be verified for operability and errors on a set of test devices. If the verification is successful, the updates are distributed to other managed devices.

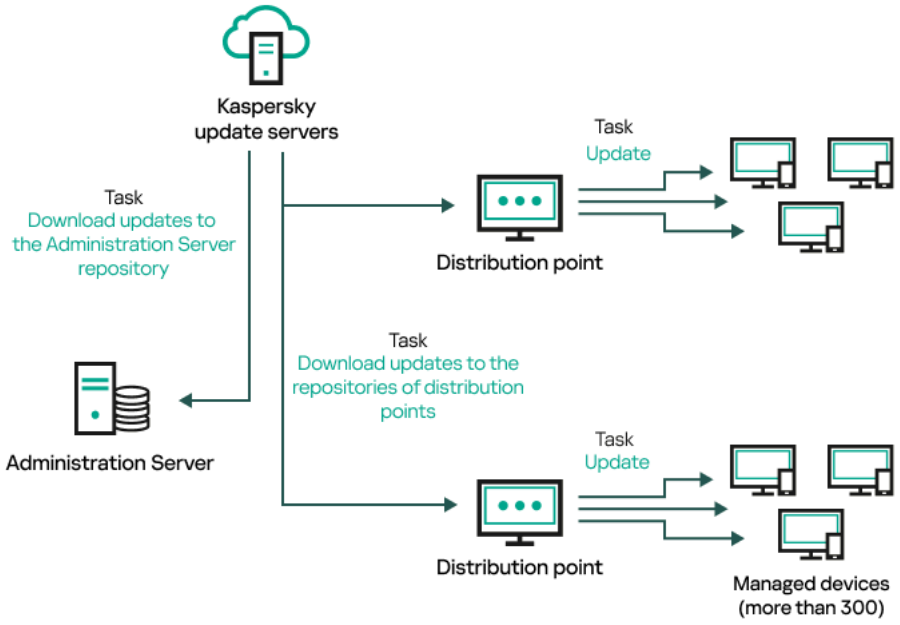
Each Kaspersky application requests required updates from Administration Server. Administration Server aggregates these requests and downloads only those updates that are requested by any application. This ensures that the same updates are not downloaded multiple times and that unnecessary updates are not downloaded at all. When running the *Download updates to the Administration Server repository* task, Administration Server sends the following information to Kaspersky update servers automatically in order to ensure the downloading of relevant versions of Kaspersky databases and software modules:

- Application ID and version
- Application setup ID
- Active key ID
- *Download updates to the repository of the Administration Server* task run ID

None of the transmitted information contains personal or other confidential data. AO Kaspersky Lab protects information in accordance with requirements established by law.

Using two tasks: the Download updates to the Administration Server repository task and the Download updates to the repositories of distribution points task

You can download updates to the repositories of distribution points directly from the Kaspersky update servers instead of the Administration Server repository, and then distribute the updates to the managed devices (see figure below). Download to the repositories of distribution points is preferable if the traffic between the Administration Server and the distribution points is more expensive than the traffic between the distribution points and Kaspersky update servers, or if your Administration Server does not have internet access.



Updating by using the *Download updates to the Administration Server repository* task and the *Download updates to the repositories of distribution points* task

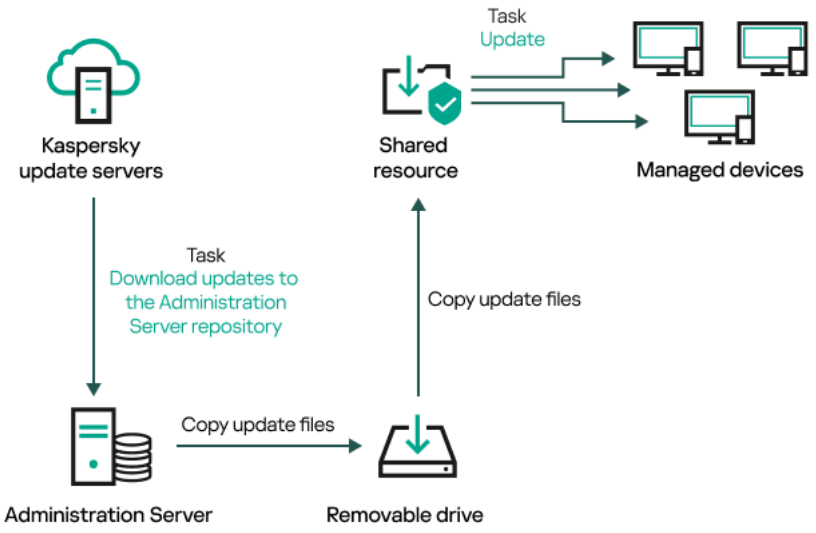
> By default, the Administration Server and distribution points communicate with Kaspersky update servers and download updates by using the HTTPS protocol. You can configure the Administration Server and/or distribution points to use the HTTP protocol instead of HTTPS.

To implement this scheme, create the *Download updates to the repositories of distribution points* task in addition to the *Download updates to the Administration Server repository* task. After that the distribution points will download updates from Kaspersky update servers, and not from the Administration Server repository.

The *Download updates to the Administration Server repository* task is also required for this scheme, because this task is used to download Kaspersky databases and software modules for Open Single Management Platform.

Manually through a shared folder or an FTP server

If the client devices do not have a connection to the Administration Server, you can use a shared resource as a source for updating Kaspersky databases, software modules, and applications. In this scheme, you need to copy required updates from the Administration Server repository to a removable drive, then copy the updates to the shared resource specified as an update source in the settings of Kaspersky Endpoint Security (see figure below).



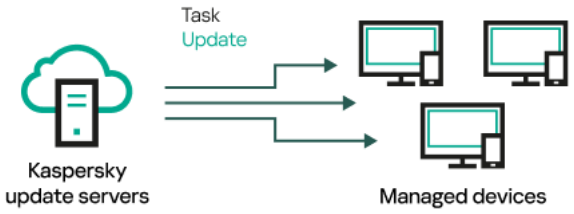
Updating through a shared folder or an FTP server

For more information about sources of updates in Kaspersky Endpoint Security, see the following Helps:

- Kaspersky Endpoint Security for Linux Help
- Kaspersky Endpoint Security for Windows Help

Directly from Kaspersky update servers to Kaspersky Endpoint Security on the managed devices

On the managed devices, you can configure Kaspersky Endpoint Security to receive updates directly from Kaspersky update servers (see figure below).



Updating security applications directly from Kaspersky update servers

In this scheme, the security application does not use the repository provided by Open Single Management Platform. To receive updates directly from Kaspersky update servers, specify Kaspersky update servers as an update source in the security application. For more information about these settings, see the following Helps:

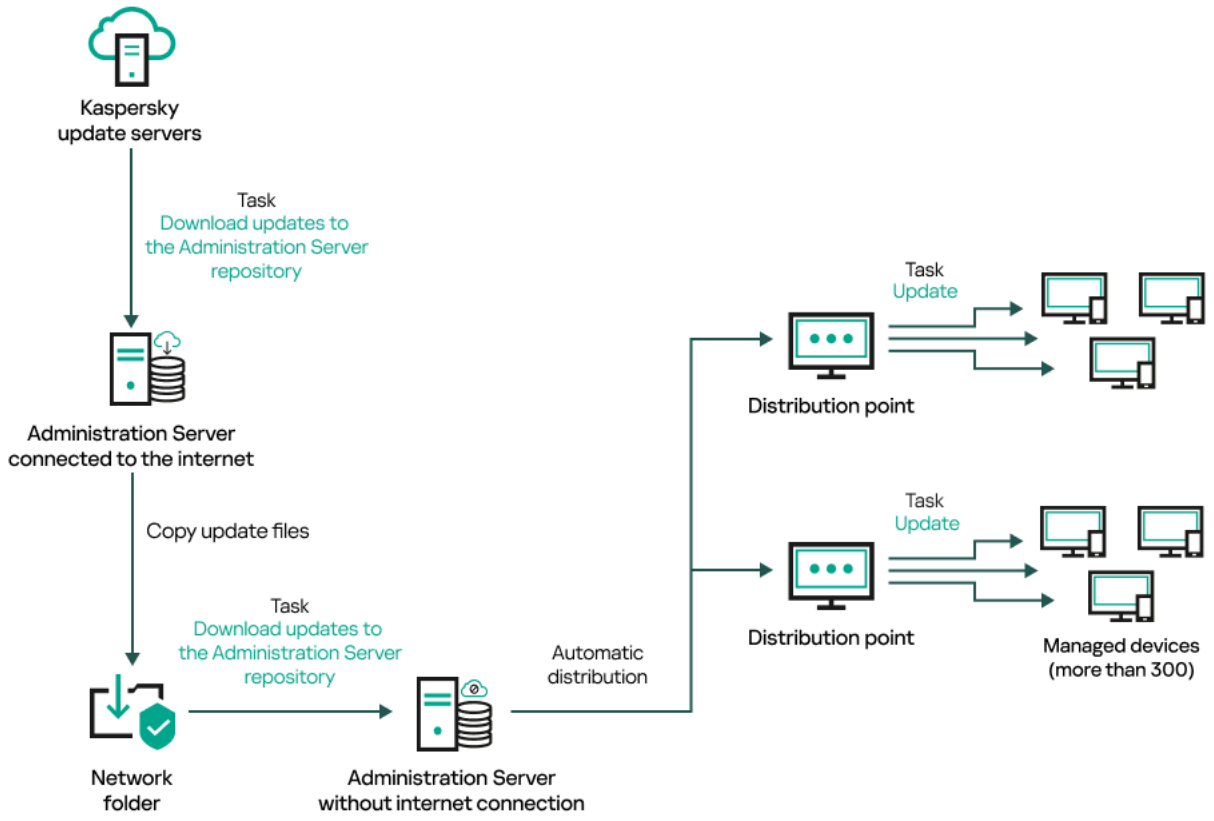
- Kaspersky Endpoint Security for Linux Help
- Kaspersky Endpoint Security for Windows Help

Through a network folder if Administration Server has no internet connection

If Administration Server has no internet connection, you can configure the *Download updates to the Administration Server repository* task to download updates from a network folder. In this case, you must copy the required update files to the specified folder from time to time. For example, you can copy the required update files from one of the following sources:

- Administration Server that has an internet connection (see the figure below)

  Because an Administration Server downloads only the updates that are requested by the security applications, the sets of security applications managed by the Administration Servers—the one that has an internet connection and the one that does not—must match.



Updating through a network folder if Administration Server has no internet connection

- Kaspersky Update Utility

  Because this utility uses the old scheme to download updates, open properties of the *Download updates to the Administration Server repository* task, and then enable the *Download updates by using the old scheme* option. If necessary, you can also configure the *Download the update request file automatically* option so that the Kaspersky Update Utility downloads only the required updates.

Creating the Download updates to the Administration Server repository task

The *Download updates to the Administration Server repository* task allows you to download updates of databases and software modules for Kaspersky security applications from Kaspersky update servers to the Administration Server repository. In the list of tasks, there can only be one *Download updates to the Administration Server repository* task.

After the *Download updates to the Administration Server repository* task is complete and the updates are downloaded, they can be propagated to the managed devices.

If you install Administration Server in a cluster, the *Download updates to the Administration Server repository* task is created automatically.

Before you distribute updates to the managed devices, you can run the *Update verification* task. This allows you to make sure that Administration Server installs the downloaded updates properly and a security level is not decreased because of the updates. To verify them before distributing, configure the **Run update verification** option in the *Download updates to the Administration Server repository* task settings.

To create a Download updates to the Administration Server repository task:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click **Add**.

   The New task wizard starts. Follow the steps of the wizard.
3. For the Open Single Management Platform application, select the **Download updates to the Administration Server repository** task type.
4. Specify the name for the task that you are creating. A task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).
5. On the **Finish task creation** page, you can enable the **Open task details when creation is complete** option to open the task properties window and modify the default task settings. Otherwise, you can configure task settings later, at any time.
6. Click the **Finish** button.

   The task is created and displayed in the list of tasks.
7. Click the created task name to open the task properties window.
8. In the task properties window, on the **Application settings** tab, specify the following settings:

   - **Sources of updates**

     As a source of updates, you can use Kaspersky update servers or a network folder. If you create a task for a secondary or virtual Administration Server, you can also select a local folder or a primary Administration Server as a source of updates.

     In the *Download updates to the Administration Server repository* task and the *Download updates to the repositories of distribution points* task, user authentication does not work if you select a password-protected local or network folder as an update source. To resolve this issue, first mount the password-protected folder, and then specify the required credentials, for example, by means of the operating system. After that, you can select this folder as an update source in an update download task. Open Single Management Platform will not require that you enter the credentials.

- **Folder for storing updates**

  The path to the specified folder for storing saved updates. You can copy the specified folder path to a clipboard. You cannot change the path to a specified folder for a group task.

- **Force update of secondary Administration Servers**

  If this option is enabled, the Administration Server starts update tasks on the secondary Administration Servers as soon as new updates are downloaded. Update tasks are started by using the source of update that is configured in the task properties on the secondary Administration Servers.

  If this option is disabled, the update tasks on the secondary Administration Servers start according to their schedules.

  By default, this option is disabled.

- **Copy downloaded updates to additional folders**

  After the Administration Server receives updates, it copies them to the specified folders. Use this option if you want to manually manage the distribution of updates on your network.

  For example, you may want to use this option in the following situation: the network of your organization consists of several independent subnets, and devices from each of the subnets do not have access to other subnets. However devices in all of the subnets have access to a common network share. In this case, you set Administration Server in one of the subnets to download updates from Kaspersky update servers, enable this option, and then specify this network share. In downloaded updates to the repository tasks for other Administration Servers, specify the same network share as the update source.

  If you select to copy the downloaded updates to a local folder, you must specify a folder on the device that has Administration Server installed. The path to the folder depends on the device's operating system. Since the local folder is located on Linux-based Administration Server, you have to specify the POSIX path, for example: /path/to/dir.

  When selecting a network folder that is located on Linux-based Administration Server, you must ensure access to the network folder in advance from the device with Administration Server installed by mounting the folder. Access to the folder is performed by using the SMB protocol.

  By default, this option is disabled.

- **Download diff files**

  This option enables the downloading diff files feature.

  By default, this option is disabled.

- **Download updates by using the old scheme**

  Open Single Management Platform downloads updates of databases and software modules by using the new scheme. For the application to download updates by using the new scheme, the update source must contain the update files with the metadata compatible with the new scheme. If the update source contains the update files with the metadata compatible with the old scheme only, enable the **Download updates by using the old scheme** option. Otherwise, the update download task will fail.

For example, you must enable this option when a local or network folder is specified as an update source and the update files in this folder were downloaded by Kaspersky Update Utility which downloads updates by using the old scheme.

By default, this option is disabled.

- **Run update verification**

  Administration Server downloads updates from the source, saves them to a temporary repository, and runs the task defined in the **Update verification task** field. If the task completes successfully, the updates are copied from the temporary repository to a shared folder on the Administration Server and then distributed to all devices for which the Administration Server acts as the source of updates (tasks with the **When new updates are downloaded to the repository** schedule type are started). The task of downloading updates to the repository is finished only after completion of the *Update verification* task.

  By default, this option is disabled.

- **Download the update request file automatically**

  If you want Kaspersky Update Utility to download only the required updates with suitable configuration, enable this option and specify the path where Kaspersky Next XDR Expert will create the update request file with information about the updates to be downloaded. The update request file contains the timestamp and the MD5 hash for version control.

  Make sure that the **Download updates by using the old scheme** option is disabled.

  Also, the update download in Kaspersky Update Utility must be configured.

  By default, the following path is specified: /var/opt/kaspersky/klnagent_srv/1093/.working/share_srv/retranslation_request.xml. If necessary, you can change the default name (retranslation_request) and path.

  The request file can be placed at a local or shared folder. When using a shared folder, you have to ensure the access to this folder from the device with Administration Server installed. The ways to ensure the access and the tools used depend on your infrastructure.

  When performing the *Download updates to the Administration Server repository* task, Kaspersky Next XDR Expert creates the update request file in the .xml format at the path that you have specified. Kaspersky Update Utility automatically obtains the information from the request file, and then downloads only the required updates. If Kaspersky Update Utility is unable to obtain the information automatically (for example, Kaspersky Update Utility does not have access to the folder where Kaspersky Next XDR Expert created the request file), you have to transmit the file by using third-party tools.

9. In the task properties window, on the **Schedule** tab, create a schedule for task start. If necessary, specify the following settings:

   - **Start task**:

     - **Manually** (selected by default)

       The task does not run automatically. You can only start it manually.

By default, this option is selected.

- **Every N minutes**

  The task runs regularly, with the specified interval in minutes, starting from the specified time on the day that the task is created.

  By default, the task runs every 30 minutes, starting from the current system time.

- **Every N hours**

  The task runs regularly, with the specified interval in hours, starting from the specified date and time.

  By default, the task runs every 6 hours, starting from the current system date and time.

- **Every N days**

  The task runs regularly, with the specified interval in days. Additionally, you can specify a date and time of the first task run. These additional options become available, if they are supported by the application for which you create the task.

  By default, the task runs every day, starting from the current system date and time.

- **Every N weeks**

  The task runs regularly, with the specified interval in weeks, on the specified day of week and at the specified time.

  By default, the task runs every Friday at the current system time.

- **Daily (daylight saving time is not supported)**

  The task runs regularly, with the specified interval in days. This schedule does not support observance of daylight saving time (DST). It means that when clocks jump one hour forward or backward at the beginning or ending of DST, the actual task start time does not change.

  We do not recommend that you use this schedule. It is needed for backward compatibility of Open Single Management Platform.

  By default, the task starts every day at the current system time.

- **Weekly**

  The task runs every week on the specified day and at the specified time.

- **By days of week**

  The task runs regularly, on the specified days of the week, at the specified time.

  By default, the task runs every Friday at 6:00:00 PM.

- **Monthly**

  The task runs regularly, on the specified day of the month, at the specified time.

  In months that lack the specified day, the task runs on the last day.

  By default, the task runs on the first day of each month, at the current system time.

- **Every month on specified days of selected weeks**

  The task runs regularly, on the specified days of each month, at the specified time.

  By default, the start time is 18:00, and no days of the month are selected.

  > Note that you select neither a specific date in the month, nor the number of the week (first, second week of the month) but the sequence number of the weekday within a month. For example, if you set the cursor in the **Tu** cell of the **First** row, it means the task will run every first Tuesday of each new month.

  You can select several days of week.

- **On completing another task**

  The current task starts after another task completes. This option only works if both tasks are assigned to the same devices. For example, you may want to run the *Manage devices* task with the **Turn on the device** option and, after it completes, run the *Virus scan* task as a triggering task.

  You have to select the triggering task from the table and the status with which this task must complete (**Completed successfully** or **Failed**).

  If necessary, you can search, sort, and filter the tasks in the table as follows:

  - Enter the task name in the search field, to search the task by its name.
  - Click the sort icon to sort the tasks by name.

    By default, the tasks are sorted in alphabetical ascending order.
  - Click the filter icon, and in the window that opens, filter the tasks by group, and then click the **Apply** button.

- Additional task settings:

  - **Run missed tasks**

    This option determines the behavior of a task if a client device is not visible on the network when the task is about to start.

    If this option is enabled, the system attempts to start the task the next time the Kaspersky application is run on the client device. If the task schedule is **Manually**, **Once** or **Immediately**, the task is started immediately after the device becomes visible on the network or immediately after the device is included in the task scope.

If this option is disabled, only scheduled tasks run on client devices. For **Manually**, **Once** and **Immediately** schedule, tasks run only on those client devices that are visible on the network. For example, you may want to disable this option for a resource-consuming task that you want to run only outside of business hours.

By default, this option is disabled.

- **Use automatically randomized delay for task starts**

  If this option is enabled, the task is started on client devices randomly within a specified time interval, that is, *distributed task start*. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  The distributed start time is calculated automatically when a task is created, depending on the number of client devices to which the task is assigned. Later, the task is always started on the calculated start time. However, when task settings are edited or the task is started manually, the calculated value of the task start time changes.

  If this option is disabled, the task starts on client devices according to the schedule.

- **Use automatically randomized delay for task starts within an interval of**

  If this option is enabled, the task is started on client devices randomly within the specified time interval. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  If this option is disabled, the task starts on client devices according to the schedule.

  By default, this option is disabled. The default time interval is one minute.

- **Stop the task if it runs longer than**

  After the specified time period expires, the task is stopped automatically, whether it is completed or not.

  Enable this option if you want to interrupt (or stop) tasks that take too long to execute.

  By default, this option is disabled. The default task execution time is 120 minutes.

10. Click the **Save** button.

    The task is created and configured.

When Administration Server performs the *Download updates to the Administration Server repository* task, updates to databases and software modules are downloaded from the updates source and stored on Administration Server. If you create this task for an administration group, it will only be applied to Network Agents included in the specified administration group.

If you use a proxy server when connecting to the internet, you have to specify the proxy server address in the Administration Server properties. Otherwise, the *Download updates to the Administration Server repository* task will not work.

Viewing downloaded updates

When Administration Server performs the *Download updates to the Administration Server repository* task, updates to databases and software modules are downloaded from the updates source and stored on Administration Server. You can view the downloaded updates in the **Updates for Kaspersky databases and software modules** section.

To view the list of downloaded updates,

In the main menu, go to **Operations** → **Kaspersky applications** → **Updates for Kaspersky databases and software modules**.

A list of available updates appears.

Verifying downloaded updates

Before installing updates to the managed devices, you can first check the updates for operability and errors through the *Update verification* task. The *Update verification* task is performed automatically as part of the *Download updates to the Administration Server repository* task. The Administration Server downloads updates from the source, saves them in the temporary repository, and runs the *Update verification* task. If the task completes successfully, the updates are copied from the temporary repository to the Administration Server repository. They are distributed to all client devices for which the Administration Server is the source of updates.

If, as a result of the *Update verification* task, updates located in the temporary repository are incorrect or if the *Update verification* task completes with an error, such updates are not copied to the Administration Server repository. The Administration Server retains the previous set of updates. Also, the tasks that have the **When new updates are downloaded to the repository** schedule type are not started then. These operations are performed at the next start of the *Download updates to the Administration Server repository* task if scanning of the new updates completes successfully.

A set of updates is considered invalid if any of the following conditions is met on at least one test device:

- An update task error occurred.
- The real-time protection status of the security application changed after the updates were applied.
- An infected object was detected during running of the on-demand scan task.
- A runtime error of a Kaspersky application occurred.

If none of the listed conditions is true for any test device, the set of updates is considered valid, and the *Update verification* task is considered to have completed successfully.

Before you start to create the *Update verification* task, perform the prerequisites:

1. Create an administration group with several test devices. You will need this group to verify the updates.

   We recommend using devices with the most reliable protection and the most popular application configuration across the network. This approach increases the quality and probability of virus detection during scans, and minimizes the risk of false positives. If viruses are detected on test devices, the *Update verification* task is considered unsuccessful.

2. Create the update and malware scan tasks for an application supported by Open Single Management Platform, for example, Kaspersky Endpoint Security for Linux. When creating the update and malware scan tasks, specify the administration group with the test devices.

   The *Update verification* task sequentially runs the update and malware scan tasks on test devices to check that all updates are valid. In addition, when creating the *Update verification* task, you need to specify the update and malware scan tasks.

3. Create the *Download updates to the Administration Server repository* task.

To make Open Single Management Platform verify downloaded updates before distributing them to client devices:

1. In the main menu, go to **Asset management** → **Tasks**.

2. Click the **Download updates to the Administration Server repository** task.

3. In the task properties window that opens, go to the **Application settings** tab, and then enable the **Run update verification** option.

4. If the *Update verification* task exists, click the **Select task** button. In the window that opens, select the *Update verification* task in the administration group with test devices.

5. If you did not create the *Update verification* task earlier, do the following:

   a. Click the **New task** button.

   b. In the New task wizard that opens, specify the task name if you want to change the preset name.

   c. Select the administration group with test devices, which you created earlier.

   d. First, select the update task of a required application supported by Open Single Management Platform, and then select the malware scan task.

      After that, the following options appear. We recommend leaving them enabled:

      - **Restart the device after database update**

        After anti-virus databases are updated on a device, we recommend rebooting the device.

        By default, the option is enabled.

      - **Check real-time protection status after database update and device restart**

        If this option is enabled, the *Update verification* task checks whether updates downloaded to the Administration Server repository are valid, and if the protection level decreased after the anti-virus database update and device restart.

        By default, this option is enabled.

   e. Specify an account from which the *Update verification* task will be run. You can use your account and leave the **Default account** option enabled. Alternatively, you can specify that the task should be run under another account that has the necessary access rights. To do this, select the **Specify account** option, and then enter the credentials of that account.

6. Click **Save** to close the properties window of the *Download updates to the Administration Server repository* task.

The automatic update verification is enabled. Now, you can run the *Download updates to the Administration Server repository* task, and it will start from update verification.

Creating the task for downloading updates to the repositories of distribution points

You can create the *Download updates to the repositories of distribution points* task for an administration group. This task will run for distribution points included in the specified administration group.

You can use this task, for example, if traffic between the Administration Server and the distribution point(s) is more expensive than traffic between the distribution point(s) and Kaspersky update servers, or if your Administration Server does not have internet access.

This task is required to download updates from Kaspersky update servers to the repositories of distribution points. The list of updates includes:

- Updates to databases and software modules for Kaspersky security applications
- Updates to Open Single Management Platform components
- Updates to Kaspersky security applications

After the updates are downloaded, they can be propagated to the managed devices.

*To create the **Download updates to the repositories of distribution points** task, for a selected administration group:*

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click the **Add** button.

   The New task wizard starts. Follow the steps of the wizard.
3. For the Open Single Management Platform application, in the **Task type** field select **Download updates to the repositories of distribution points**.
4. Specify the name for the task that you are creating. A task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).
5. Select an option button to specify the administration group, the device selection, or the devices to which the task applies.
6. At the **Finish task creation** step, if you want to modify the default task settings, enable the **Open task details when creation is complete** option. If you do not enable this option, the task is created with the default settings. You can modify the default settings later, at any time.
7. Click the **Create** button.

   The task is created and displayed in the list of tasks.
8. Click the name of the created task to open the task properties window.

9. On the **Application settings** tab of the task properties window, specify the following settings:

- **Sources of updates**

  The following resources can be used as a source of updates for the distribution point:

  - Kaspersky update servers

    HTTP(S) servers at Kaspersky from which Kaspersky applications download database and application module updates.

    This option is selected by default.

  - Local or network folder

    A local or network folder that contains the latest updates. Only a mounted SMB share can be used as a network folder. When selecting a local folder, you must specify a folder on the device that has Administration Server installed.

    In the *Download updates to the Administration Server repository* task and the *Download updates to the repositories of distribution points* task, user authentication does not work if you select a password-protected local or network folder as an update source. To resolve this issue, first mount the password-protected folder, and then specify the required credentials, for example, by means of the operating system. After that, you can select this folder as an update source in an update download task. Open Single Management Platform will not require that you enter the credentials.

- **Folder for storing updates**

  The path to the specified folder for storing saved updates. You can copy the specified folder path to a clipboard. You cannot change the path to a specified folder for a group task.

- **Download diff files**

  This option enables the downloading diff files feature.

  By default, this option is disabled.

- **Download updates by using the old scheme**

  Open Single Management Platform downloads updates of databases and software modules by using the new scheme. For the application to download updates by using the new scheme, the update source must contain the update files with the metadata compatible with the new scheme. If the update source contains the update files with the metadata compatible with the old scheme only, enable the **Download updates by using the old scheme** option. Otherwise, the update download task will fail.

  For example, you must enable this option when a local or network folder is specified as an update source and the update files in this folder were downloaded by Kaspersky Update Utility which downloads updates by using the old scheme.

  By default, this option is disabled.

10. Create a schedule for task start. If necessary, specify the following settings:

- **Start task**:

  - **Manually** (selected by default)

    The task does not run automatically. You can only start it manually.

    By default, this option is selected.

  - **Every N minutes**

    The task runs regularly, with the specified interval in minutes, starting from the specified time on the day that the task is created.

    By default, the task runs every 30 minutes, starting from the current system time.

  - **Every N hours**

    The task runs regularly, with the specified interval in hours, starting from the specified date and time.

    By default, the task runs every 6 hours, starting from the current system date and time.

  - **Every N days**

    The task runs regularly, with the specified interval in days. Additionally, you can specify a date and time of the first task run. These additional options become available, if they are supported by the application for which you create the task.

    By default, the task runs every day, starting from the current system date and time.

  - **Every N weeks**

    The task runs regularly, with the specified interval in weeks, on the specified day of week and at the specified time.

    By default, the task runs every Friday at the current system time.

  - **Daily (daylight saving time is not supported)**

    The task runs regularly, with the specified interval in days. This schedule does not support observance of daylight saving time (DST). It means that when clocks jump one hour forward or backward at the beginning or ending of DST, the actual task start time does not change.

    We do not recommend that you use this schedule. It is needed for backward compatibility of Open Single Management Platform.

    By default, the task starts every day at the current system time.

  - **Weekly**

    The task runs every week on the specified day and at the specified time.

- **By days of week**

  The task runs regularly, on the specified days of the week, at the specified time.

  By default, the task runs every Friday at 6:00:00 PM.

- **Monthly**

  The task runs regularly, on the specified day of the month, at the specified time.

  In months that lack the specified day, the task runs on the last day.

  By default, the task runs on the first day of each month, at the current system time.

- **Every month on specified days of selected weeks**

  The task runs regularly, on the specified days of each month, at the specified time.

  By default, the start time is 18:00, and no days of the month are selected.

  > Note that you select neither a specific date in the month, nor the number of the week (first, second week of the month) but the sequence number of the weekday within a month. For example, if you set the cursor in the **Tu** cell of the **First** row, it means the task will run every first Tuesday of each new month.

  You can select several days of week.

- **On virus outbreak**

  The task runs after a *Virus outbreak* event occurs. Select application types that will monitor virus outbreaks. The following application types are available:

  - Anti-virus for workstations and file servers
  - Anti-virus for perimeter defense
  - Anti-virus for mail systems

  By default, all application types are selected.

  You may want to run different tasks depending on the security application type that reports a virus outbreak. In this case, remove the selection of the application types that you do not need.

- **On completing another task**

  The current task starts after another task completes. This option only works if both tasks are assigned to the same devices. For example, you may want to run the *Manage devices* task with the **Turn on the device** option and, after it completes, run the *Virus scan* task as a triggering task.

  You have to select the triggering task from the table and the status with which this task must complete (**Completed successfully** or **Failed**).

If necessary, you can search, sort, and filter the tasks in the table as follows:

- Enter the task name in the search field, to search the task by its name.
- Click the sort icon to sort the tasks by name.

  By default, the tasks are sorted in alphabetical ascending order.
- Click the filter icon, and in the window that opens, filter the tasks by group, and then click the **Apply** button.

- **Run missed tasks**

  This option determines the behavior of a task if a client device is not visible on the network when the task is about to start.

  If this option is enabled, the system attempts to start the task the next time the Kaspersky application is run on the client device. If the task schedule is **Manually**, **Once** or **Immediately**, the task is started immediately after the device becomes visible on the network or immediately after the device is included in the task scope.

  If this option is disabled, only scheduled tasks run on client devices. For **Manually**, **Once** and **Immediately** schedule, tasks run only on those client devices that are visible on the network. For example, you may want to disable this option for a resource-consuming task that you want to run only outside of business hours.

  By default, this option is disabled.

- **Use automatically randomized delay for task starts**

  If this option is enabled, the task is started on client devices randomly within a specified time interval, that is, *distributed task start*. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  The distributed start time is calculated automatically when a task is created, depending on the number of client devices to which the task is assigned. Later, the task is always started on the calculated start time. However, when task settings are edited or the task is started manually, the calculated value of the task start time changes.

  If this option is disabled, the task starts on client devices according to the schedule.

- **Use automatically randomized delay for task starts within an interval of**

  If this option is enabled, the task is started on client devices randomly within the specified time interval. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  If this option is disabled, the task starts on client devices according to the schedule.

  By default, this option is disabled. The default time interval is one minute.

11. Click the **Save** button.

The task is created and configured.

In addition to the settings that you specify during task creation, you can change other properties of a created task.

When the *Download updates to the repositories of distribution points* task is performed, updates for databases and software modules are downloaded from the update source and stored in the distribution points repository. Downloaded updates will only be used by distribution points that are included in the specified administration group and that have no update download task explicitly set for them.

Adding sources of updates for the Download updates to the Administration Server repository task

When you create or use the task for downloading updates to the Administration Server repository, you can choose the following sources of updates:

- Kaspersky update servers
- Primary Administration Server

  This resource applies to tasks created for a secondary or virtual Administration Server.
- Local or network folder

  This resource applies to tasks created for a secondary or virtual Administration Server.
- Network folder

> In the *Download updates to the Administration Server repository* task and the *Download updates to the repositories of distribution points* task, user authentication does not work if you select a password-protected local or network folder as an update source. To resolve this issue, first mount the password-protected folder, and then specify the required credentials, for example, by means of the operating system. After that, you can select this folder as an update source in an update download task. Open Single Management Platform will not require that you enter the credentials.

Kaspersky update servers are used by default, but you can also download updates from a local or network folder. You might want to use the folder if your network does not have access to the internet. In this case, you can manually download updates from Kaspersky update servers and put the downloaded files in the necessary folder.

You can specify only one path to a local or network folder. As a local folder, you must specify a folder on the device where Administration Server is installed. As a network folder, you can use an FTP or HTTP server or an SMB share. If an SMB share requires authentication, it must be mounted in the system with the required credentials in advance. We recommend not using the SMB1 protocol since it is insecure.

If you add both Kaspersky update servers and the local or network folder, updates will be downloaded first from the folder. In the case of an error when downloading, Kaspersky update servers will be used.

In case a shared folder that contains updates is password-protected, enable the **Specify account for access to shared folder of the update source (if any)** option and enter the account credentials required for access.

To add the sources of updates:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click **Download updates to the Administration Server repository**.
3. Go to the **Application settings** tab.

4. On the **Sources of updates** line, click the **Configure** button.

5. In the window that opens, click the **Add** button.

6. In the update source list, add the necessary sources. If you select the **Network folder** or **Local or network folder** check box, specify a path to the folder.

7. Click **OK**, and then close the update source properties window.

8. In the update source window, click **OK**.

9. Click the **Save** button in the task window.

Now updates are downloaded to the Administration Server repository from the specified sources.

Approving and declining software updates

The settings of an update installation task may require approval of updates that are to be installed. You can approve updates that must be installed and decline updates that must not be installed.

For example, you may want to first check the installation of updates in a test environment and make sure that they do not interfere with the operation of devices, and only then allow the installation of these updates on client devices.

> Approving and declining updates is available only for Network Agent and managed applications installed on the Windows-based client devices. Seamless updating of Administration Server, OSMP Console, and management web plug-ins is not supported. To update these components, you have to download the latest versions from the Kaspersky website, and then install them manually.

To approve or decline one or several updates:

1. In the main menu, go to **Operations** → **Kaspersky applications** → **Seamless updates**.

   A list of available updates appears.

   > Updates of managed applications may require a specific minimum version of Kaspersky Security Center to be installed. If this version is later than your current version, these updates are displayed but cannot be approved. Also, no installation packages can be created from such updates until you upgrade Kaspersky Security Center. You are prompted to upgrade your Kaspersky Security Center instance to the required minimum version.

2. If necessary, accept EULA by clicking the **View and accept License Agreements** button.

3. Select the updates that you want to approve or decline.

4. Click **Approve** to approve the selected updates or **Decline** to decline the selected updates.

   The default value is *Undefined*.

The updates to which you assign *Approved* status are placed in a queue for installation.

The updates to which you assign *Declined* status are uninstalled (if possible) from all devices on which they were previously installed. Also, they will not be installed on other devices in future.

Some updates for Kaspersky applications cannot be uninstalled. If you set *Declined* status for them, Open Single Management Platform will not uninstall these updates from the devices on which they were previously installed. However, these updates will never be installed on other devices in future.

If you set *Declined* status for third-party software updates, these updates will not be installed on devices for which they were planned but have not yet been installed. Updates will remain on devices on which they were already installed. If you have to delete the updates, you can manually delete them locally.

Automatic installation of updates for Kaspersky Endpoint Security for Windows

You can configure automatic updates of databases and software modules of Kaspersky Endpoint Security for Windows on client devices.

To configure download and automatic installation of updates of Kaspersky Endpoint Security for Windows on devices:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click the **Add** button.

   The New task wizard starts. Follow the steps of the wizard.
3. For the Kaspersky Endpoint Security for Windows application, select **Update** as the task subtype.
4. Specify the name for the task that you are creating. A task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).
5. Choose the task scope.
6. Specify the administration group, the device selection, or the devices to which the task applies.
7. At the **Finish task creation** step, if you want to modify the default task settings, enable the **Open task details when creation is complete** option. If you do not enable this option, the task is created with the default settings. You can modify the default settings later, at any time.
8. Click the **Create** button.

   The task is created and displayed in the list of tasks.
9. Click the name of the created task to open the task properties window.

10. On the **Application settings** tab of the task properties window, define the update task settings in local or mobile mode:

    - **Local mode**: Connection is established between the device and the Administration Server.

    - **Mobile mode**: No connection is established between Open Single Management Platform and the device (for example, when the device is not connected to the internet).

11. Enable the update sources that you want to use to update databases and application modules for Kaspersky Endpoint Security for Windows. If required, change positions of the sources in the list by using the **Move up** and **Move down** buttons. If several update sources are enabled, Kaspersky Endpoint Security for Windows tries to connect to them one after another, starting from the top of the list, and performs the update task by retrieving the update package from the first available source.

12. Enable the **Install approved application module updates** option to download and install software module updates together with the application databases.

    If the option is enabled, Kaspersky Endpoint Security for Windows notifies the user about available software module updates and includes software module updates in the update package when running the update task. Kaspersky Endpoint Security for Windows installs only those updates for which you have set the *Approved* status; they will be installed locally through the application interface or through Open Single Management Platform.

    You can also enable the **Automatically install critical application module updates** option. If any updates are available for software modules, Kaspersky Endpoint Security for Windows automatically installs those that have *Critical* status; the remaining updates will be installed after you approve them.

    If updating the software module requires reviewing and accepting the terms of the License Agreement and Privacy Policy, the application installs updates after the terms of the License Agreement and Privacy Policy have been accepted by the user.

13. Select the **Copy updates to folder** check box in order for the application to save downloaded updates to a folder, and then specify the folder path.

14. Schedule the task. To ensure timely updates, we recommend that you select the **When new updates are downloaded to the repository** option.

15. Click **Save**.

When the **Update** task is running, the application sends requests to Kaspersky update servers.

Some updates require installation of the latest versions of management plug-ins.

About using diff files for updating Kaspersky databases and software modules

When Open Single Management Platform downloads updates from Kaspersky update servers, it optimizes traffic by using diff files. You can also enable the usage of diff files by devices (Administration Servers, distribution points, and client devices) that take updates from other devices on your network.

About the Downloading diff files feature

A diff file describes the differences between two versions of a file of a database or software module. The usage of diff files saves traffic inside your company's network because diff files occupy less space than entire files of databases and software modules. If the *Downloading diff files* feature is enabled on Administration Server or a distribution point, the diff files are saved on this Administration Server or distribution point. As a result, devices that take updates from this Administration Server or distribution point can use the saved diff files to update their databases and software modules.

To optimize the usage of diff files, we recommend that you synchronize the update schedule of devices with the update schedule of the Administration Server or distribution point from which the devices take updates. However, the traffic can be saved even if devices are updated several times less often than are the Administration Server or distribution point from which the devices take updates.

Distribution points do not use IP multicasting for automatic distribution of diff files.

Enabling the Downloading diff files feature

Stages

1. **Enabling the feature on Administration Server**

   Enable the feature in the settings of a *Download updates to the repository of the Administration Server* task.

2. **Enabling the feature for a distribution point**

   Enable the feature for a distribution point that receives updates by means of a *Download updates to the repositories of distribution points* task.

   Then enable the feature in the Network Agent policy settings for a distribution point that receives updates from Administration Server.

   Then enable the feature for a distribution point that receives updates from Administration Server.

   The feature is enabled in the Network Agent policy settings and—if the distribution points are assigned manually and if you want to override policy settings—in the **Distribution points** section of the Administration Server properties.

To check that the Downloading diff files feature is successfully enabled, you can measure the internal traffic before and after you perform the scenario.

Downloading updates by distribution points

Open Single Management Platform allows distribution points to receive updates from the Administration Server, Kaspersky servers, or from a local or network folder.

To configure update download for a distribution point:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Distribution points** section.

3. Click the name of the distribution point through which updates will be delivered to client devices in the group.

4. In the distribution point properties window, select the **Source of updates** section.

5. Select an update source for the distribution point:

- **Source of updates**

  Select a source of updates for the distribution point:

  - To allow the distribution point to receive updates from the Administration Server, select **Retrieve from Administration Server**.
  - To allow the distribution point to receive updates by using a task, select **Use update download task**, and then specify a *Download updates to the repositories of distribution points* task:
    - If such a task already exists on the device, select the task in the list.
    - If no such task yet exists on the device, click the **Create task** link to create a task. The New task wizard starts. Follow the instructions of the wizard.

- **Download diff files**

  This option enables the downloading diff files feature.

  By default, this option is enabled.

The distribution point will receive updates from the specified source.

Updating Kaspersky databases and software modules on offline devices

Updating Kaspersky databases and software modules on managed devices is an important task for maintaining protection of the devices against viruses and other threats. Administrators usually configure regular updates through usage of the Administration Server repository.

When you need to update databases and software modules on a device (or a group of devices) that is not connected to the Administration Server (primary or secondary), a distribution point or the internet, you have to use alternative sources of updates, such as an FTP server or a local folder. In this case, you have to deliver the files of the required updates by using a mass storage device, such as a flash drive or an external hard drive.

You can copy the required updates from:

- The Administration Server.

  To be sure the Administration Server repository contains the updates required for the security application installed on an offline device, at least one of the managed online devices must have the same security application installed. This application must be configured to receive the updates from the Administration Server repository through the *Download updates to the Administration Server repository* task.

- Any device that has the same security application installed and configured to receive the updates from the Administration Server repository, a distribution point repository, or directly from the Kaspersky update servers.

Below is an example of configuring updates of databases and software modules by copying them from the Administration Server repository.

To update Kaspersky databases and software modules on offline devices:

1. Connect the removable drive to the device where the Administration Server is installed.

2. Copy the updates files to the removable drive.

   By default, the updates are located at: /var/opt/kaspersky/klnagent_srv/1093/.working/share_srv/Updates/.

   Alternatively, you can configure Open Single Management Platform to regularly copy the updates to the folder that you select. For this purpose, use the **Copy downloaded updates to additional folders** option in the properties of the *Download updates to the Administration Server repository* task. If you specify a folder located on a flash drive or an external hard drive as a destination folder for this option, this mass storage device will always contain the latest version of the updates.

3. On offline devices, configure Kaspersky Endpoint Security to receive updates from a local folder or a shared resource, such as an FTP server or a shared folder.

   How-to instructions:

   - Kaspersky Endpoint Security for Linux Help
   - Kaspersky Endpoint Security for Windows Help

4. Copy the updates files from the removable drive to the local folder or the shared resource that you want to use as an update source.

5. On the offline device that requires update installation, start the *Update* task of Kaspersky Endpoint Security for Linux or Kaspersky Endpoint Security for Windows, depending on the operating system of the offline device.

After the update task is complete, the Kaspersky databases and software modules are up-to-date on the device.

Remote diagnostics of client devices

You can use remote diagnostics for remote execution of the following operations on Windows-based and Linux-based client devices:

- Enabling and disabling tracing, changing the tracing level, and downloading the trace file

- Downloading system information and application settings

- Downloading event logs
- Generating a dump file for an application
- Starting diagnostics and downloading diagnostics reports
- Starting, stopping, and restarting applications

You can use event logs and diagnostics reports downloaded from a client device to troubleshoot problems on your own. Also, if you contact Kaspersky Technical Support, a Technical Support specialist might ask you to download trace files, dump files, event logs, and diagnostics reports from a client device for further analysis at Kaspersky.

Opening the remote diagnostics window

To perform remote diagnostics on Windows-based and Linux-based client devices, you first have to open the remote diagnostics window.

To open the remote diagnostics window:

1. To select the device for which you want to open the remote diagnostics window, perform one of the following:
   - If the device belongs to an administration group, in the main menu, go to **Asset management** → **Assets (Devices)**.
   - If the device belongs to the Unassigned devices group, in the main menu, go to **Discovery & deployment** → **Unassigned devices**.
2. Click the name of the required device.
3. In the device properties window that opens, select the **Advanced** tab.
4. On the **Advanced** tab, click **Remote diagnostics**.

   This opens the **Remote diagnostics** window of a client device. If connection between Administration Server and the client device is not established, the error message displays.

Alternatively, if you need to obtain all diagnostic information about a Linux-based client device at once, you can run the collect.sh script on this device.

Enabling and disabling tracing for applications

You can enable and disable tracing for applications, including Xperf tracing.

Enabling and disabling tracing

To enable or disable tracing on a remote device:

1. Open the remote diagnostics window of a client device.

2. In the remote diagnostics window, select the **Kaspersky applications** tab.

   In the **Application management** section, the list of Kaspersky applications installed on the device displays.

3. In the list of applications, select the application for which you want to enable or disable tracing.

   The list of remote diagnostics options opens.

4. If you want to enable tracing:

   a. In the **Tracing** section, click **Enable tracing**.

   b. In the **Modify tracing level** window that opens, we recommend that you keep the default values of the settings. When required, a Technical Support specialist will guide you through the configuration process. The following settings are available:

   - **Tracing level**

     The tracing level defines the amount of detail that the trace file contains.

   - **Rotation-based tracing**

     The application overwrites the tracing information to prevent excessive increase in the size of the trace file. Specify the maximum number of files to be used to store the tracing information, and the maximum size of each file. If the maximum number of trace files of the maximum size are written, the oldest trace file is deleted so that a new trace file can be written.

     This setting is available for Kaspersky Endpoint Security only.

   c. Click **Save**.

   The tracing is enabled for the selected application. In some cases, the security application and its task must be restarted in order to enable tracing.

   On Linux-based client devices, tracing for the Updater of Network Agent component is regulated by the Network Agent settings. Therefore, the **Enable tracing** and **Modify tracing level** options are disabled for this component on client devices running Linux.

5. If you want to disable tracing for the selected application, click the **Disable tracing** button.

   The tracing is disabled for the selected application.

Enabling Xperf tracing

For Kaspersky Endpoint Security, a Technical Support specialist may ask you to enable Xperf tracing for information about the system performance.

To enable and configure Xperf tracing or disable it:

1. Open the remote diagnostics window of a client device.

2. In the remote diagnostics window, select the **Kaspersky applications** tab.

   In the **Application management** section, the list of Kaspersky applications installed on the device displays.

3. In the list of applications, select Kaspersky Endpoint Security for Windows.

   The list of remote diagnostics options for Kaspersky Endpoint Security for Windows displays.

4. In the **Xperf tracing** section, click **Enable Xperf tracing**.

   If Xperf tracing is already enabled, the **Disable Xperf tracing** button is displayed instead. Click this button if you want to disable Xperf tracing for Kaspersky Endpoint Security for Windows.

5. In the **Change Xperf tracing level** window that opens, depending on the request from the Technical Support specialist, do the following:

   a. Select one of the following tracing levels:

   - **Light level**

     A trace file of this type contains the minimum amount of information about the system.

     By default, this option is selected.

   - **Deep level**

     A trace file of this type contains more detailed information than trace files of the *Light* type and may be requested by Technical Support specialists when a trace file of the *Light* type is not enough for the performance evaluation. A *Deep* trace file contains technical information about the system including information about hardware, operating system, list of started and finished processes and applications, events used for performance evaluation, and events from Windows System Assessment Tool.

   b. Select one of the following Xperf tracing types:

   - **Basic type**

     The tracing information is received during operation of the Kaspersky Endpoint Security application.

     By default, this option is selected.

   - **On-restart type**

     The tracing information is received when the operating system starts on the managed device. This tracing type is effective when the issue that affects the system performance occurs after the device is turned on and before Kaspersky Endpoint Security starts.

   You may also be asked to enable the **Rotation file size, in MB** option to prevent excessive increase in the size of the trace file. Then specify the maximum size of the trace file. When the file reaches the maximum size, the oldest tracing information is overwritten with new information.

c. Define the rotation file size.

d. Click **Save**.

Xperf tracing is enabled and configured.

6. If you want to disable Xperf tracing for Kaspersky Endpoint Security for Windows, click **Disable Xperf tracing** in the **Xperf tracing** section.

   Xperf tracing is disabled.

Downloading trace files of an application

To download a trace file of an application:

1. Open the remote diagnostics window of a client device.

2. In the remote diagnostics window, select the **Kaspersky applications** tab.

   In the **Application management** section, the list of Kaspersky applications installed on the device displays.

3. In the list of applications, select the application for which you want to download a trace file.

4. In the **Tracing** section, click the **Trace files** button.

   This opens the **Device tracing logs** window, where a list of trace files is displayed.

5. In the list of trace files, select the file that you want to download.

6. Do one of the following:

   - Download the selected file by clicking **Download**. You can select one or several files for downloading.

   - Download a portion of the selected file:

     a. Click **Download a portion**.

     You cannot download portions of several files at the same time. If you select more than one trace file, the **Download a portion** button will be disabled.

     b. In the window that opens, specify the name and the file portion to download, according to your needs.

     For Linux-based devices, editing the file portion name is not available.

     c. Click **Download**.

The selected file, or its portion, is downloaded to the location that you specify.

Deleting trace files

You can delete trace files that are no longer needed.

To delete a trace file:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window that opens, select the **Event logs** tab.
3. In the **Trace files** section, click **Windows Update logs** or **Remote installation logs**, depending on which trace files you want to delete.

   The **Windows Update logs** link is available only for Windows-based client devices.

   This opens the **Device tracing logs** window, where a list of trace files is displayed.
4. In the list of trace files, select one or several files that you want to delete.
5. Click the **Remove** button.

The selected trace files are deleted.

Downloading application settings

To download application settings from a client device:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window, select the **Kaspersky applications** tab.
3. In the **Application settings** section, click the **Download** button to download information about the settings of the applications installed on the client device.

The ZIP archive with information is downloaded to the specified location.

Downloading system information from a client device

To download system information from a client device:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window, select the **System information** tab.
3. Click the **Download** button to download the system information about the client device.

   If you obtain system information about a Linux-based device, a dump file for emergency terminated applications is added to the resulting file.

The file with information is downloaded to the specified location.

Downloading event logs

To download an event log from a remote device:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window, on the **Event logs** tab, click **All device logs**.
3. In the **All device logs** window, select one or several relevant logs.
4. Do one of the following:
   - Download the selected log by clicking **Download entire file**.
   - Download a portion of the selected log:
     a. Click **Download a portion**.
     You cannot download portions of several logs at the same time. If you select more than one event log, the **Download a portion** button will be disabled.
     b. In the window that opens, specify the name and the log portion to download, according to your needs.
     For Linux-based devices, editing the log portion name is not available.
     c. Click **Download**.

The selected event log, or a portion of it, is downloaded to the specified location.

Starting, stopping, restarting the application

You can start, stop, and restart applications on a client device.

To start, stop, or restart an application:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window, select the **Kaspersky applications** tab.
   In the **Application management** section, the list of Kaspersky applications installed on the device displays.
3. In the list of applications, select the application that you want to start, stop, or restart.
4. Select an action by clicking one of the following buttons:
   - **Stop application**
     This button is available only if the application is currently running.
   - **Restart application**
     This button is available only if the application is currently running.

- **Start application**

  This button is available only if the application is not currently running.

Depending on the action that you have selected, the required application is started, stopped, or restarted on the client device.

If you restart the Network Agent, a message is displayed stating that the current connection of the device to the Administration Server will be lost.

Running the remote diagnostics of Kaspersky Security Center Network Agent and downloading the results

To start diagnostics for Kaspersky Security Center Network Agent on a remote device and download the results:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window, select the **Kaspersky applications** tab.

   In the **Application management** section, the list of Kaspersky applications installed on the device displays.
3. In the list of applications, select **Kaspersky Security Center Network Agent**.

   The list of remote diagnostics options opens.
4. In the **Diagnostics report** section, click the **Run diagnostics** button.

   This starts the remote diagnostics process and generates a diagnostics report. When the diagnostics process is complete, the **Download diagnostics report** button becomes available.
5. Click the **Download diagnostics report** button to download the report.

The report is downloaded to the specified location.

Running an application on a client device

You may have to run an application on a client device if a Kaspersky support specialist requests it. You do not have to install the application on that device.

If you want to run a custom script on the client device, you can use the `KLACDT_SAVE_SETTING` environment variable to determine the path where the execution results will be saved. For example:

- For a Windows-based device:

  ```
  @echo some result > %KLACDT_SAVE_SETTING%\res.log
  ```

- For a Linux-based device:

  ```
  echo some result > $KLACDT_SAVE_SETTING/res.log
  ```

To run an application on the client device:

1. Open the remote diagnostics window of a client device.
2. In the remote diagnostics window, select the **Running a remote application** tab.

3. In the **Object type** field, select the type of object that you want to upload:

   - Select **File** to upload the executable file.

   - Select **Folder** to upload the folder containing the application that you want to run on the client device.

     If you select this object type, the **Name of the executable file to be run** field appears. In this field, specify the name of the executable file in the folder. You can also specify the relative path to the file.

   - Select **Archive** to upload the ZIP archive containing the application that you want to run on the client device.

     If you select this object type, the **Name of the executable file to be run** field appears. In this field, specify the name of the executable file in the folder. You can also specify the relative path to the file.

     > The ZIP archive must include the utility folder. This folder contains the executable file to be run on a remote device.

4. Upload the object by selecting it in the system window, or dragging it to the upload field.

5. If necessary, in the **Command line arguments** field, specify the command line arguments to be passed when running the executable file.

6. Click the **Upload and run** button to run the selected application on a client device.

7. After the application execution is completed successfully, download the execution results by clicking the **Download application execution results** button.

   > The application execution results are stored until you run a new diagnostic or close the window.

Remote diagnostics of the client device by using the application is completed.

Generating a dump file for an application

An application dump file allows you to view the parameters of the application running on a client device at a point in time. This file also contains information about modules that were loaded for an application.

> Obtaining dump files from Linux-based devices is not supported.

To obtain dump files through remote diagnostics, the kldumper utility is used. This utility is designed to obtain the dump files of processes of Kaspersky applications at the request of technical support specialists. Detailed information on the requirements for using the kldumper utility is provided in the Open Single Management Platform Knowledge Base.

To create a dump file for an application:

1. Open the remote diagnostics window of a client device.

2. In the remote diagnostics window, select the **Running a remote application** tab.

3. In the **Generating the process dump file** section, specify the executable file of the application for which you want to generate a dump file.

4. Click the **Download dump file** button.

   An archive with the dump file for the specified application is downloaded.

   If the specified application is not running on the client device, the "result" folder contained in the downloaded archive will be empty.

   If the specified application is running, but the downloading fails with an error or the "result" folder contained in the downloaded archive is empty, refer to the Open Single Management Platform Knowledge Base.

Running remote diagnostics on a Linux-based client device

Open Single Management Platform allows you to download the basic diagnostic information from a client device. Alternatively, you can obtain the diagnostic information about a Linux-based device by using the collect.sh script by Kaspersky. This script is run on the Linux-based client device that needs to be diagnosed, and then it generates a file with the diagnostic information, the system information about this device, trace files of applications, device logs, and a dump file for emergency-terminated applications.

We recommend that you use the collect.sh script to obtain all diagnostic information about the Linux-based client device at once. If you download the diagnostic information remotely through Open Single Management Platform, you will need to go through all sections of the remote diagnostics interface. Also the diagnostic information for a Linux-based device will probably not be obtained completely.

If you need to send the generated file with the diagnostic information to the Kaspersky Technical Support, delete all confidential information before sending the file.

To download the diagnostic information from a Linux-based client device by using the collect.sh script:

1. Download the collect.sh script packed in the collect.tar.gz archive.

2. Copy the downloaded archive to the Linux-based client device that needs to be diagnosed.

3. Run the following command to unpack the collect.tar.gz archive:

   ```
   # tar -xzf collect.tar.gz
   ```

4. Run the following command to specify the script execution rights:

   ```
   # chmod +x collect.sh
   ```

5. Run the collect.sh script by using an account with administrator rights:

   ```
   # ./collect.sh
   ```

A file with the diagnostic information is generated and saved to the /tmp/$HOST_NAME-collect.tar.gz folder.

Managing third-party applications and executable files on client devices

This section describes the features of Open Single Management Platform related to the management of third-party applications and executable files run on client devices.

Using Application Control to manage executable files

You can use the Application Control component to allow or block startup of executable files on user devices. The Application Control component supports Windows-based and Linux-based operating systems.

> For Linux-based operating systems, Application Control component is available starting from Kaspersky Endpoint Security 11.2 for Linux.

Prerequisites

- Open Single Management Platform is deployed in your organization.
- The policy of Kaspersky Endpoint Security for Linux or Kaspersky Endpoint Security for Windows is created and is active. The Application Control component is enabled in the policy.

Stages

The Application Control usage scenario proceeds in stages:

1. **Forming and viewing the list of executable files on client devices**

   This stage helps you find out what executable files are found on managed devices. View the list of executable files and compare it with the lists of allowed and prohibited executable files. The restrictions on executable files usage can be related to the information security polices in your organization.

   How-to instructions: Obtaining and viewing a list of executable files stored on client devices

2. **Creating categories for executable files used in your organization**

   Analyze the lists of executable files stored on managed devices. Based on the analysis, create categories for executable files. It is recommended to create a "Work applications" category that covers the standard set of executable files that are used at your organization. If different security groups use their own sets of executable files in their work, a separate category can be created for each security group.

   Startup of executable files whose settings do not match any of the Application Control rules is regulated by the selected operating mode of the component:

   - *Denylist*. The mode is used if you want to allow the startup of all executable files except those specified in block rules. This mode is selected by default.
   - *Allowlist*. The mode is used if you want to block the startup of all executable files except those specified in allow rules.

The Application Control rules are implemented through categories for executable files. In Open Single Management Platform there are three types of categories for executable files:

- Category with content added manually. You define conditions, for example, file metadata, file hashcode, file certificate, file path, to include executable files in the category.
- Category that includes executable files from selected devices. You specify a device whose executable files are automatically included in the category.
- Category that includes executable files from selected folder. You specify a folder from which executable files are automatically included in the category.

3. **Configuring Application Control in the Kaspersky Endpoint Security policy**

   Configure the Application Control component in Kaspersky Endpoint Security for Linux policy using the categories you have created on the previous stage.

   How-to instructions: Configuring Application Control in the Kaspersky Endpoint Security for Windows policy

4. **Turning on Application Control component in test mode**

   To ensure that Application Control rules do not block executable files required for user's work, it is recommended to enable testing of Application Control rules and analyze their operation after creating new rules. When testing is enabled, Kaspersky Endpoint Security for Windows will not block executable files whose startup is forbidden by Application Control rules, but will instead send notifications about their startup to the Administration Server.

   When testing Application Control rules, it is recommended to perform the following actions:

   - Determine the testing period. Testing period can vary from several days to two months.
   - Examine the events resulting from testing the operation of Application Control.

   How-to instructions for OSMP Console: Configuring Application Control component in the Kaspersky Endpoint Security for Windows policy. Follow this instruction and enable the **Test Mode** option in configuration process.

5. **Changing the settings of Application Control component**

   If necessary, make changes to the Application Control settings. Based on the test results, you can add executable files related to events of the Application Control component to a category with content added manually.

   How-to instructions: OSMP Console: Adding event-related executable files to the application category

6. **Applying the rules of Application Control in operation mode**

   After Application Control rules are tested and configuration of categories is complete, you can apply the rules of Application Control in operation mode.

   How-to instructions for OSMP Console: Configuring Application Control component in the Kaspersky Endpoint Security for Windows policy. Follow this instruction and disable the **Test Mode** option in configuration process.

7. **Verifying Application Control configuration**

   Be sure that you have done the following:

   - Created categories for executable files.
   - Configured Application Control using the categories.
   - Applied the rules of Application Control in operation mode.

Results

When the scenario is complete, startup of executable files on managed devices is controlled. The users can run only those executable files that are allowed in your organization and cannot run executable files that are prohibited in your organization.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Application Control modes and categories

The Application Control component monitors users' attempts to start executable files. You can use Application Control rules to control the startup of executable files.

> Application Control component is available for Kaspersky Endpoint Security 11.2 for Linux and later versions.

Startup of executable files whose settings do not match any of the Application Control rules is regulated by the selected operating mode of the component:

- *Denylist*. The mode is used if you want to allow the startup of all executable files except those specified in block rules. This mode is selected by default.
- *Allowlist*. The mode is used if you want to block the startup of all executable files except those specified in allow rules.

The Application Control rules are implemented through categories for executable files. In Open Single Management Platform there are three types of categories:

- Category with content added manually. You define conditions, for example, file metadata, file hashcode, file certificate, file path, to include executable files in the category.
- Category that includes executable files from selected devices. You specify a device whose executable files are automatically included in the category.
- Category that includes executable files from selected folder. You specify a folder from which executable files are automatically included in the category.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Obtaining and viewing a list of applications installed on client devices

Open Single Management Platform inventories all software installed on managed client devices running Linux and Windows.

Network Agent compiles a list of applications installed on a device, and then transmits this list to Administration Server. It takes about 10-15 minutes for the Network Agent to update the application list.

For Windows-based client devices, Network Agent receives most of the information about installed applications from the Windows registry. For Linux-based client devices, package managers provide information about installed applications to Network Agent.

To view the list of applications installed on managed devices:

1. In the main menu, go to **Operations** → **Third-party applications** → **Applications registry**.

   The page displays a table with the applications that are installed on managed devices. Select the application to view its properties, for example, vendor name, version number, list of executable files, list of devices on which the application is installed, list of available software updates, and list of detected software vulnerabilities.

2. You can group and filter the data of the table with installed applications as follows:

   - Click the settings icon (⚙) in the upper-right corner of the table.

     In the invoked **Table settings** menu, select the columns to be displayed in the table. To view the operating system type of the client devices on which the application is installed, select the **OS type** column.

   - Click the filter icon (▽) in the upper-right corner of the table, and then specify and apply the filter criterion in the invoked menu.

     The filtered table of installed applications is displayed.

To view the list of applications installed on a specific managed device,

In the main menu, go to **Devices** → **Managed devices** → **device name** → **Advanced** → **Applications registry**. In this menu, you can export the list of applications to a CSV file or TXT file.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Obtaining and viewing a list of executable files stored on client devices

You can obtain the list of executable files stored on client devices in one of the following ways:

- Enabling notifications about applications startup in Kaspersky Endpoint Security policy.
- Creating an inventory task.

Enabling notifications about applications startup in Kaspersky Endpoint Security policy

To enable notifications about applications startup:

1. Open the Kaspersky Endpoint Security policy settings, and then go to **General settings** → **Reports and Storage**.

2. In the **Data transfer to Administration Server** settings group, select the **About started applications** check box, and save the changes.

When a user attempts to start executable files, information about these files is added to the list of executable files on a client device. Kaspersky Endpoint Security sends this information to Network Agent, and then Network Agent sends it to Administration Server.

Creating an inventory task

For Kaspersky Endpoint Security for Linux, the feature of inventorying executable files is available since no earlier that version 11.2.

You can reduce load on the database while obtaining information about the installed applications. To save database space, run an inventory task on reference devices on which a standard set of software is installed. The preferable number of devices is 1-3.

To create an inventory task for executable files on client devices:

1. In the main menu, go to **Asset management** → **Tasks**.

   The list of tasks is displayed.

2. Click the **Add** button.

   The New task wizard starts. Follow the steps of the wizard.

3. On the **New task settings** page, from the **Application** drop-down list, select Kaspersky Endpoint Security for Linux or Kaspersky Endpoint Security for Windows, depending on the operating system of the client devices.

4. From the **Task type** drop-down list, select **Inventory**.

5. On the **Finish task creation** page, click the **Finish** button.

   After the New task wizard has finished, the **Inventory** task is created and configured. If you want, you can change the settings for the created task. The newly created task is displayed in the list of tasks.

For a detailed description of the inventory task, see the Kaspersky Endpoint Security for Linux Help and the Kaspersky Endpoint Security for Windows Help.

After the **Inventory** task is performed, the list of executable files stored on managed devices is formed, and you can view the list.

During inventory, executable files in the following formats can be detected (depending on the option that you select in the inventory task properties): MZ, COM, PE, NE, SYS, CMD, BAT, PS1, JS, VBS, REG, MSI, CPL, DLL, JAR, and HTML.

Viewing the list of executable files stored on managed devices

To view the list of executable files stored on client devices:

In the main menu, go to **Operations** → **Third-party applications** → **Executable files**.

The page displays the list of executable files stored on client devices.

If necessary, you can send the executable file of the managed device to the device where your OSMP Console is open.

To send an executable file:

1. In the main menu, go to **Operations** → **Third-party applications** → **Executable files**.
2. Click the link of the executable file that you want to send.
3. In the window that opens, go to the **Devices** section, and then select the check box of the managed device from which you want to send the executable file.

   Before you send the executable file, make sure that the managed device has a direct connection to the Administration Server, by selecting the **Do not disconnect from the Administration Server** check box.

4. Click the **Send** button.

The selected executable file is downloaded for further sending to the device where your OSMP Console is open.

Creating an application category with content added manually

You can specify a set of criteria as a template of executable files for which you want to allow or block a start in your organization. On the basis of executable files corresponding to the criteria, you can create an application category and use it in the Application Control component configuration.

To create an application category with content added manually:

1. In the main menu, go to **Operations** → **Third-party applications** → **Application categories**.

   The page with a list of application categories is displayed.

2. Click the **Add** button.

   The New category wizard starts. Proceed through the wizard by using the **Next** button.

3. On the **Select category creation method** step, specify the application category name and select the **Category with content added manually. Data of executable files is manually added to the category** option.
4. On the **Conditions** step, click the **Add** button to add a condition criterion to include files in the creating category.
5. On the **Condition criteria** step, do the following:

   a. Select a rule type for the creation of the category:

   - **From KL category**

     If this option is selected, you can specify a Kaspersky application category as the condition of adding applications to the user category. The applications from the specified Kaspersky category will be added to the user application category.

   - **Select certificate from repository**

     If this option is selected, you can specify certificates from the repository. The category condition matches only the executable files signed by the specified certificate.

Certificates are added to the repository if you created and started an inventory task or selected the **About started applications** check box in the Kaspersky Endpoint Security policy. You can perform both actions.

- **Specify path to application (masks supported)**

  If this option is selected, you can specify the path to the folder on the client device containing the executable files that are to be added to the user application category.

- **Removable drive**

  If this option is selected, you can specify the type of the medium (any drive or removable drive) on which the application is run. Applications that have been run on the selected drive type are added to the user application category.

- **Hash, metadata, or certificate**:

  - **Select from list of executable files**

    If this option is selected, you can use the list of executable files on the client device to select and add applications to the category.

  - **Select from applications registry**

    If this option is selected, application registry is displayed. After you select an application from the registry, the window opens with the parameters filled in with metadata from the application that you selected:

    - File name.
    - File version. You can specify precise value of the version or describe a condition, for example "greater than 5.0".
    - Application name.
    - Application version. You can specify precise value of the version or describe a condition, for example "greater than 5.0".
    - Vendor.

    Note that only the launch of executable files that meet the specified parameters is blocked, not the launch of the application you select. If the selected application metadata matches the one of the executable file that is launched when you launch the application, then you can proceed to the next step. Otherwise, you have to change the values manually to match the metadata of the executable file.

  - **Specify manually**

    If this option is selected, you must specify file hash, or metadata, or certificate as the condition of adding applications to the user category.

    File Hash

Depending on the version of the security application installed on devices on your network, you should select an algorithm for hash value computing by Open Single Management Platform for files in this category. Information about hash values computed by hash functions is stored in the Administration Server database.

SHA256 is a cryptographic hash function: no vulnerabilities have been found in its algorithm, and so it is considered the most reliable cryptographic function nowadays. Kaspersky Endpoint Security for Linux supports SHA256 computing.

Select either of the options of hash value computing by Open Single Management Platform for files in the category:

- If all instances of security applications installed on your network are Kaspersky Endpoint Security for Linux, select the **SHA256** check box.
- Select the **MD5 hash** check box only if you use Kaspersky Endpoint Security for Windows. Kaspersky Endpoint Security for Linux does not support the MD5 hash function.

Metadata

If this option is selected, you can specify file metadata as file name, file version, vendor. The category condition matches only the executable files with the same metadata.

Certificate

If this option is selected, you can specify certificates from the repository. The category condition matches only the executable files signed by the specified certificate.

Certificates are added to the repository if you created and started an inventory task or selected the **About started applications** check box in the Kaspersky Endpoint Security policy. You can perform both actions.

- **From archived folder**

If this option is selected, you can specify a file of an archived folder, and then select which condition you want to use to add applications to the user category. The archived folder is unpacked and the conditions that you select are applied to the files in the folder. As a condition, you can select one of the following criteria:

- **File Hash**

  You select which hash function (MD5 or SHA256) you want to use to calculate hash values. The applications that have the same hash value as the files in the archived folder are added to the user application category.

  Select an MD5 hash function only if you use Kaspersky Endpoint Security for Windows. Kaspersky Endpoint Security for Linux does not support the MD5 hash function.

- **Metadata**

  You select which metadata you want to use as criteria. Executable files that contain the same metadata will be added to the user application category.

- **Certificate**

  You select which certificate properties (certificate subject, fingerprint, or issuer) you want to use as criteria. Executable files that have been signed with the certificates that have the same properties will be added to the user category.

Metadata and **Certificate** criteria must be specified manually after uploading a file or an archived folder if you want to use these criteria.

If this option is selected, you can specify a file of an archived folder, and then select which condition you want to use to add applications to the user category. The archived folder is unpacked and the conditions that you select are applied to the files in the folder. As a condition, you can select one of the following criteria:

- **File Hash**

  You select which hash function (MD5 or SHA256) you want to use to calculate hash values. The applications that have the same hash value as the files in the archived folder are added to the user application category.

  Select an MD5 hash function only if you use Kaspersky Endpoint Security for Windows. Kaspersky Endpoint Security for Linux does not support the MD5 hash function.

- **Metadata**

  You select which metadata you want to use as criteria. Executable files that contain the same metadata will be added to the user application category.

- **Certificate**

  You select which certificate properties (certificate subject, fingerprint, or issuer) you want to use as criteria. Executable files that have been signed with the certificates that have the same properties will be added to the user category.

Metadata and **Certificate** criteria must be specified manually after uploading a file or an archived folder if you want to use these criteria.

b. If necessary, add a comment in the **Description** field. The number of symbols cannot exceed 1000.

Later, you can edit or delete the comment by opening the **Conditions** tab in the application category properties, and then in the **Condition criterion** column, click the link with the condition for which you want to edit or delete the comment.

When selecting the **From KL category** rule type, you can add one comment for all conditions to be created based on this rule type. You can edit or delete the comment for every condition separately.

The selected criterion is added to the list of conditions.

You can add as many criteria as you need for the application category that is being created.

6. On the **Exclusions** step, click the **Add** button to add an exclusive condition criterion to exclude files from the category that is being created.

   If necessary, you can add a comment in the **Description** field. The number of symbols cannot exceed 1000.

   Later, you can edit or delete the comment by opening the **Exclusions** tab in the application category properties, and then in the **Condition criterion** column, click the link with the condition for which you want to edit or delete the comment.

7. On the **Condition criteria** step, select a rule type from the list, in the same way that you selected a rule type for category creation.

When the wizard finishes, the application category is created and displayed in the list of application categories. You can use the created application category when you configure Application Control.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Creating an application category that includes executable files from selected devices

You can use executable files from selected devices as a template of executable files that you want to allow or block. Based on executable files from selected devices, you can create an application category and use it in the Application Control component configuration.

Make sure that the following prerequisites are met:

- The Application Control component is enabled in the Kaspersky Endpoint Security policy.
- A list of executable files stored on managed devices has been obtained.

To create application category that includes executable files from selected devices:

1. In the main menu, go to **Operations** → **Third-party applications** → **Application categories**.

   The page with a list of categories of executable files is displayed.

2. Click the **Add** button.

   The New category wizard starts. Proceed through the wizard by using the **Next** button.

3. On the **Select category creation method** step, specify the category name and select the **Category that includes executable files from selected devices. These executable files are processed automatically and their metrics are added to the category** option.

4. Click **Add**.

5. In the window that opens, select a device or devices whose executable files will be used to create the application category.

6. Specify the following settings:

   - **Hash value computing algorithm**

     Depending on the version of the security application installed on devices on your network, you should select an algorithm for hash value computing by Open Single Management Platform for files in this category. Information about hash values computed by hash functions is stored in the Administration Server database.

     SHA256 is a cryptographic hash function: no vulnerabilities have been found in its algorithm, and so it is considered the most reliable cryptographic function nowadays. Kaspersky Endpoint Security for Linux supports SHA256 computing.

     Select either of the options of hash value computing by Open Single Management Platform for files in the category:

     - If all instances of security applications installed on your network are Kaspersky Endpoint Security for Linux, select the **SHA256** check box.

     Select the **MD5 hash** check box only if you use Kaspersky Endpoint Security for Windows. Kaspersky Endpoint Security for Linux does not support the MD5 hash function.

The **Calculate SHA256 for files in this category (supported by Kaspersky Endpoint Security 10 Service Pack 2 for Windows and any later versions)** check box is selected by default.

The **Calculate MD5 for files in this category (supported by versions earlier than Kaspersky Endpoint Security 10 Service Pack 2 for Windows)** is cleared by default.

- **Synchronize data with Administration Server repository**

  Select this option if you want that Administration Server periodically to check changes in the specified folder (or folders).

  By default, this option is disabled.

  If you enable this option, specify the period (in hours) to check changes in the specified folder (folders). By default, scan interval is 24 hours.

- **File type**

  In this section, you can specify file type that is used to create the application category.

  All files. All files are taken into consideration when creating the category. By default, this option is selected.

  Only files outside the application categories. Only files outside the application categories are taken into consideration when creating the category.

- **Folders**

  In this section you can specify which folders from the selected device (devices) contain files that are used to create the application category.

  All folders. All folders are taken into consideration for the creating category. By default, this option is selected.

  Specified folder. Only specified folder is taken into consideration for the creating category. If you select this option you must specify path to the folder.

When the wizard finishes, the category of executable files is created. It is displayed in the list of categories. You can use the created category when you configure Application Control.

Creating an application category that includes executable files from selected folder

You can use executable files from a selected folder as a standard of executable files that you want to allow or block in your organization. On the basis of executable files from the selected folder, you can create an application category and use it in the Application Control component configuration.

To create a category that includes executable files from the selected folder:

1. In the main menu, go to **Operations** → **Third-party applications** → **Application categories**.

   The page with a list of categories is displayed.

2. Click the **Add** button.

   The New category wizard starts. Proceed through the wizard by using the **Next** button.

3. On the **Select category creation method** step, specify the category name and select the **Category that includes executable files from a specific folder. Executable files of applications copied to the specified folder are automatically processed and their metrics are added to the category** option.

4. Specify the folder whose executable files will be used to create the category.

5. Define the following settings:

   - **Include dynamic-link libraries (DLL) in this category**

     The application category includes dynamic-link libraries (files in DLL format), and the Application Control component logs the actions of such libraries running in the system. Including DLL files in the category may lower the performance of Open Single Management Platform.

     By default, this check box is cleared.

   - **Include script data in this category**

     The application category includes data on scripts, and scripts are not blocked by Web Threat Protection. Including the script data in the category may lower the performance of Open Single Management Platform.

     By default, this check box is cleared.

   - **Hash value computing algorithm**: **Calculate SHA256 for files in this category (supported by Kaspersky Endpoint Security 10 Service Pack 2 for Windows and later versions)** / **Calculate MD5 for files in this category (supported by versions earlier than Kaspersky Endpoint Security 10 Service Pack 2 for Windows)**

     Depending on the version of the security application installed on devices on your network, you should select an algorithm for hash value computing by Open Single Management Platform for files in this category. Information about hash values computed by hash functions is stored in the Administration Server database.

     SHA256 is a cryptographic hash function: no vulnerabilities have been found in its algorithm, and so it is considered the most reliable cryptographic function nowadays. Kaspersky Endpoint Security for Linux supports SHA256 computing.

     Select either of the options of hash value computing by Open Single Management Platform for files in the category:

     - If all instances of security applications installed on your network are Kaspersky Endpoint Security for Linux, select the **SHA256** check box.

     Select the **MD5 hash** check box only if you use Kaspersky Endpoint Security for Windows. Kaspersky Endpoint Security for Linux does not support the MD5 hash function.

     The **Calculate SHA256 for files in this category (supported by Kaspersky Endpoint Security 10 Service Pack 2 for Windows and any later versions)** check box is selected by default.

     The **Calculate MD5 for files in this category (supported by versions earlier than Kaspersky Endpoint Security 10 Service Pack 2 for Windows)** is cleared by default.

- **Force folder scan for changes**

  If this option is enabled, the application regularly checks the folder of category content addition for changes. You can specify the frequency of checks (in hours) in the entry field next to the check box. By default, the time interval between forced checks is 24 hours.

  If this option is disabled, the application does not force any checks of the folder. The Server attempts to access files if they have been modified, added, or deleted.

  By default, this option is disabled.

When the wizard finishes, the category of executable files is created. It is displayed in the list of categories. You can use the category at Application Control configuration.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Viewing the list of application categories

You can view the list of configured categories of executable files and the settings of each category.

To view the list of application categories,

In the main menu, go to **Operations** → **Third-party applications** → **Application categories**.

The page with a list of categories is displayed.

To view properties of an application category,

Click the name of the category.

The properties window of the category is displayed. The properties are grouped on several tabs.

Configuring Application Control in the Kaspersky Endpoint Security for Windows policy

After you create Application Control categories, you can use them for configuring Application Control in Kaspersky Endpoint Security for Windows policies.

To configure Application Control in the Kaspersky Endpoint Security for Windows policy:

1. In the main menu, go to **Asset management** → **Policies**.

   A page with a list of policies is displayed.

2. Click the **Kaspersky Endpoint Security for Windows** policy.

   The policy settings window opens.

3. Go to **Application settings** → **Security Controls** → **Application Control**.

   The **Application Control** window with Application Control settings is displayed.

4. The **Application Control** option is enabled by default. Switch the toggle button **Application Control DISABLED** to disable the option.

5. In the **Application Control Settings** block settings, enable the operation mode to apply the Application Control rules and allow Kaspersky Endpoint Security for Windows to block startup of applications.

   If you want to test the Application Control rules, in the **Application Control Settings** section, enable the test mode. In the test mode, Kaspersky Endpoint Security for Windows does not block startup of applications, but logs information about triggered rules in the report. Click the **View report** link to view this information.

6. Enable the **Control DLL modules load** option if you want Kaspersky Endpoint Security for Windows to monitor the loading of DLL modules when applications are started by users.

   Information about the module and the application that loaded the module will be saved to a report.

   Kaspersky Endpoint Security for Windows monitors only the DLL modules and drivers loaded after the **Control DLL modules load** option is selected. Restart the device after selecting the **Control DLL modules load** option if you want Kaspersky Endpoint Security for Windows to monitor all DLL modules and drivers, including those loaded before Kaspersky Endpoint Security for Windows is started.

7. (Optional) In the **Message templates** block, change the template of the message that is displayed when an application is blocked from starting and the template of the email message that is sent to you.

8. In the **Application Control Mode** block settings, select the **Denylist** or **Allowlist** mode.

   By default, the **Denylist** mode is selected.

9. Click the **Rules Lists Settings** link.

   The **Denylists and allowlists** window opens to let you add an application category. By default, the **Denylist** tab is selected if the **Denylist** mode is selected, and the **Allowlist** tab is selected if the **Allowlist** mode is selected.

10. In the **Denylists and allowlists** window, click the **Add** button.

    The **Application Control rule** window opens.

11. Click the **Please choose a category** link.

    The **Application Category** window opens.

12. Add the application category (or categories) that you created earlier.

    You can edit the settings of a created category by clicking the **Edit** button.

    You can create a new category by clicking the **Add** button.

    You can delete a category from the list by clicking the **Delete** button.

13. After the list of application categories is complete, click the **OK** button.

    The **Application Category** window closes.

14. In the **Application Control** rule window, in the **Subjects and their rights** section, create a list of users and groups of users to apply the Application Control rule.

15. Click the **OK** button to save the settings and to close the **Application Control rule** window.

16. Click the **OK** button to save the settings and to close the **Denylists and allowlists** window.

17. Click the **OK** button to save the settings and to close the **Application Control** window.

18. Close the window with the Kaspersky Endpoint Security for Windows policy settings.

    Application Control is configured. After the policy is propagated to the client devices, the startup of executable files is managed.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Adding event-related executable files to the application category

After you configure Application Control in the Kaspersky Endpoint Security policies, the following events will be displayed in the list of events:

- **Application startup prohibited** (*Critical* event). This event is displayed if you have configured Application Control to apply rules.
- **Application startup prohibited in test mode** (*Info* event). This event is displayed if you have configured Application Control to test rules.
- **Message to administrator about application startup prohibition** (*Warning* event). This event is displayed if you have configured Application Control to apply rules and a user has requested access to the application that is blocked at startup.

It is recommended to create event selections to view events related to Application Control operation.

You can add executable files related to Application Control events to an existing application category or to a new application category. You can add executable files only to an application category with content added manually.

To add executable files related to Application Control events to an application category:

1. In the main menu, go to **Monitoring** → **Event selections**.

   The list of event selections is displayed.

2. Select the event selection to view events related to Application Control and start this event selection.

   If you have not created event selection related to Application Control, you can select and start a predefined selection, for example, **Recent events**.

   The list of events is displayed.

3. Select the events whose associated executable files you want to add to the application category, and then click the **Assign to category** button.

   The New category wizard starts. Proceed through the wizard by using the **Next** button.

4. On the wizard page, specify the relevant settings:

   - In the **Action on executable file related to the event** section, select one of the following options:

     - **Add to a new application category**

       Select this option if you want to create a new application category based on event-related executable files.

By default, this option is selected.

If you have selected this option, specify a new category name.

- **Add to an existing application category**

  Select this option if you want to add event-related executable files to an existing application category.

  By default, this option is not selected.

  If you have selected this option, select the application category with content added manually to which you want to add executable files.

- In the **Rule type** section, select one of the following options:

  - **Rules for adding to inclusions**

  - **Rules for adding to exclusions**

- In the **Parameter used as a condition** section, select one of the following options:

  - **Certificate details (or SHA256 hashes for files without a certificate)**

    Files may be signed with a certificate. Multiple files may be signed with the same certificate. For example, different versions of the same application may be signed with the same certificate, or several different applications from the same vendor may be signed with the same certificate. When you select a certificate, several versions of an application or several applications from the same vendor may end up in the category.

    Each file has its own unique SHA256 hash function. When you select an SHA256 hash function, only one corresponding file, for example, the defined application version, ends up in the category.

    Select this option if you want to add to the category rules the certificate details of an executable file (or the SHA256 hash function for files without a certificate).

    By default, this option is selected.

  - **Certificate details (files without a certificate will be skipped)**

    Files may be signed with a certificate. Multiple files may be signed with the same certificate. For example, different versions of the same application may be signed with the same certificate, or several different applications from the same vendor may be signed with the same certificate. When you select a certificate, several versions of an application or several applications from the same vendor may end up in the category.

    Select this option if you want to add the certificate details of an executable file to the category rules. If the executable file has no certificate, this file will be skipped. No information about this file will be added to the category.

  - **Only SHA256 (files without a hash will be skipped)**

    Each file has its own unique SHA256 hash function. When you select an SHA256 hash function, only one corresponding file, for example, the defined application version, ends up in the category.

    Select this option if you want to add only the details of the SHA256 hash function of the executable file.

- **Only MD5 (discontinued mode, only for Kaspersky Endpoint Security 10 Service Pack 1 version)**

  Select this option only if you use Kaspersky Endpoint Security for Windows. Kaspersky Endpoint Security for Linux does not support an MD5 hash function.

  Each file has its own unique MD5 hash function. When you select an MD5 hash function, only one corresponding file, for example, the defined application version, ends up in the category.

5. Click **OK**.

   When the wizard finishes, executable files related to the Application Control events are added to the existing application category or to a new application category. You can view settings of the application category that you have modified or created.

For detailed information about Application Control, refer to the Kaspersky Endpoint Security for Linux Help and Kaspersky Endpoint Security for Windows Help.

Creating the Install required updates and fix vulnerabilities task

The *Install required updates and fix vulnerabilities* task is only available under the Vulnerability and patch management license.

The *Install required updates and fix vulnerabilities* task is used to update and fix vulnerabilities in the third-party software installed on managed devices. This task allows you to install multiple updates and fix multiple vulnerabilities according to the rules, which you specify in the task settings.

To install updates or fix vulnerabilities by using the *Install required updates and fix vulnerabilities* task, you can do one of the following:

- Run the Update installation wizard or the Vulnerability fix wizard.
- Create an *Install required updates and fix vulnerabilities* task.
- Add a rule for update installation to an existing *Install required updates and fix vulnerabilities* task.

To create the Install required updates and fix vulnerabilities task:

1. In the main menu, go to **Asset management** → **Tasks**.
2. Click **Add**.

   The New task wizard starts. Proceed through the wizard by using the **Next** button.
3. In the **Application** drop-down list, select Open Single Management Platform.
4. In the **Task type** list, select the **Install required updates and fix vulnerabilities** task type.

   If the task is not displayed, make sure that your account has the **Read**, **Write**, and **Execute** rights for the **System management**: **Vulnerability and patch management** functional area. You cannot create and configure the *Install required updates and fix vulnerabilities* task without these access rights.

5. In the **Task name** field, specify the name of the new task.

   The task name cannot be more than 100 characters long and cannot include any special characters ("*<>?\:|).

6. Select the devices to which the task will be assigned.

7. At the **Specify rules for installing updates** step of the wizard, add rules for update installation.

   > These rules are applied to installation of updates on client devices. If rules are not specified, the task has nothing to perform. For information about operations with rules, refer to Rules for update installation.

   These rules are applied to the installation of updates on client devices. If you do not specify any rules, the task has nothing to perform.

8. Specify the following settings:

   - **Start installation at device restart or shutdown**

     > If this option is enabled, updates are installed when the device is restarted or shut down. Otherwise, updates are installed according to a schedule.
     >
     > Use this option if installing the updates might affect the device performance.
     >
     > By default, this option is disabled.

   - **Install the required general system components**

     > If this option is enabled, before installing an update the application automatically installs all general system components (prerequisites) that are required to install the update. For example, these prerequisites can be operating system updates.
     >
     > If this option is disabled, you may have to install the prerequisites manually.
     >
     > By default, this option is disabled.

   - **Allow installation of new application versions during updates**

     > If this option is enabled, updates are allowed when they result in installation of a new version of a software application.
     >
     > If this option is disabled, the software is not upgraded. You can then install new versions of the software manually or through another task. For example, you may use this option if your company infrastructure is not supported by a new software version or if you want to check an upgrade in a test infrastructure.
     >
     > By default, this option is enabled.
     >
     > > Upgrading an application may cause malfunction of dependent applications installed on client devices.

   - **Download updates to the device without installing them**

     > If this option is enabled, the application downloads updates to the device but does not install them automatically. You can then Install downloaded updates manually.

Microsoft updates are downloaded to the system Windows storage. Updates of third-party applications (applications made by software vendors other than Kaspersky and Microsoft) are downloaded to the folder specified in the **Download updates to** field.

If this option is disabled, the updates are installed to the device automatically.

By default, this option is disabled.

- **Download updates to**

  This folder is used to download updates of third-party applications (applications made by software vendors other than Kaspersky and Microsoft).

- **Enable advanced diagnostics**

  If this feature is enabled, Network Agent writes trace files even if writing trace files is disabled for Network Agent in Open Single Management Platform Remote Diagnostics Utility. Traces are written to two files in turn; the total size of both files is determined by the **Maximum size, in MB, of advanced diagnostics files** value. When both files are full, Network Agent starts writing to them again. The files with traces are stored in the %WINDIR%\Temp folder. These files are accessible in the remote diagnostics utility, you can download or delete them there.

  If this feature is disabled, Network Agent writes traces according to the settings in Open Single Management Platform Remote Diagnostics Utility. No additional traces are written.

  When creating a task, you do not have to enable advanced diagnostics. You may want to use this feature later if, for example, a task run fails on some of the devices and you want to get additional information during another task run.

  By default, this option is disabled.

- **Maximum size, in MB, of advanced diagnostics files**

  The default value is 100 MB, and available values are between 1 MB and 2048 MB. You may be asked to change the default value by Kaspersky Technical Support specialists when information in the advanced diagnostics files sent by you is not enough to troubleshoot the problem.

Go to the next step of the wizard.

9. Specify the operating system restart settings:

- **Do not restart the device**

  Client devices are not restarted automatically after the operation. To complete the operation, you must restart a device (for example, manually or through a device management task). Information about the required restart is saved in the task results and in the device status. This option is suitable for tasks on servers and other devices where continuous operation is critical.

- **Restart the device**

  Client devices are always restarted automatically if a restart is required for completion of the operation. This option is useful for tasks on devices that provide for regular pauses in their operation (shutdown or restart).

- **Prompt user for action**

  The restart reminder is displayed on the screen of the client device, prompting the user to restart it manually. Some advanced settings can be defined for this option: text of the message for the user, the message display frequency, and the time interval after which a restart will be forced (without the user's confirmation). This option is most suitable for workstations where users must be able to select the most convenient time for a restart.

  By default, this option is selected.

- **Repeat prompt every (min)**

  If this option is enabled, the application prompts the user to restart the operating system with the specified frequency.

  By default, this option is enabled. The default interval is 5 minutes. Available values are between 1 and 1440 minutes.

  If this option is disabled, the prompt is displayed only once.

- **Restart after (min)**

  After prompting the user, the application forces restart of the operating system upon expiration of the specified time interval.

  By default, this option is enabled. The default delay is 30 minutes. Available values are between 1 and 1440 minutes.

- **Wait time before forced closure of applications in blocked sessions (min)**

  Applications are forced to close when the user's device goes locked (automatically after a specified interval of inactivity, or manually).

  If this option is enabled, applications are forced to close on the locked device upon expiration of the time interval specified in the entry field.

  If this option is disabled, applications do not close on the locked device.

  By default, this option is disabled.

10. At the **Finish task creation** step of the wizard, enable the **Open task details when creation is complete** option to modify the default task settings.

    If you do not enable this option, the task will be created with the default settings. You can modify the default settings later.

11. Click the **Finish** button.

    The New task wizard creates the task. If you enabled the **Open task details when creation is complete** option, the task properties window automatically opens. In this window, you can specify the general task settings and, if required, change the settings specified during task creation.

    You can also open the task properties window by clicking the name of the created task in the list of tasks.

    The task is created, configured, and displayed in the list of tasks.

12. To run the task, select it in the task list, and then click the **Start** button.

You can also set a task start schedule on the **Schedule** tab of the task properties window.

For a detailed description of scheduled start settings, refer to the general task settings.

After the task is completed, the required updates are installed and the vulnerabilities are fixed.

If in the network of your organization the WSUS server is used (local WSUS server), and you want to use it together with Kaspersky Security Center Administration Server, you must first ensure that Kaspersky Security Center Administration Server is not used as WSUS server, and then do the following:

a. On your managed devices running Windows, specify your local WSUS server for Windows Update service. You can do it either manually in Windows Registry, or by using Windows group policy.

b. On your local WSUS server, specify the required classifications for updates and the applications on which the updates are to be distributed.

c. Synchronize your WSUS server with the upstream server (this can be either a Microsoft WSUS server, or another local upstream WSUS-server).

After the *Install required updates and fix vulnerabilities* task is run on the device, Network Agent uses the Windows Update service to install Windows updates. Windows Update service obtains the address of the external WSUS server from the registry and downloads the updates directly from that external server.

Find vulnerabilities and required updates task settings

In addition to the general task settings, you can specify the following settings when creating the *Find vulnerabilities and required updates* task or later, when configuring the properties of the created task:

- **Search for vulnerabilities and updates listed by Microsoft**

  When searching for vulnerabilities and updates, Open Single Management Platform uses the information about applicable Microsoft updates from the source of Microsoft updates, which are available at the present moment.

  For example, you may want to disable this option if you have different tasks with different settings for Microsoft updates and updates of third-party applications.

  By default, this option is enabled.

  Information about optional Microsoft Windows updates is not being sent to the Administration Server.

- **Connect to the update server to update data**

  Windows Update Agent on a managed device connects to the source of Microsoft updates. The following servers can act as a source of Microsoft updates:

  - Windows Server with Microsoft Windows Server Update Services (WSUS) deployed in your organization's network
  - Microsoft Updates servers

  If this option is enabled, Windows Update Agent on a managed device connects to the source of Microsoft updates to refresh the information about applicable Microsoft Windows updates.

  If this option is disabled, Windows Update Agent on a managed device uses the information about applicable Microsoft Windows updates that was received from the source of Microsoft updates earlier.

Connecting to the source of Microsoft updates can be resource-consuming. You might want to disable this option if you set regular connection to this source of updates in another task or in the properties of Network Agent policy, in the section **Software updates and vulnerabilities**. If you do not want to disable this option, then, to reduce the Server overload, you can configure the task schedule to randomize delay for task starts within 360 minutes.

By default, this option is enabled.

Combination of the following options of the settings of Network Agent policy defines the mode of getting updates:

- Windows Update Agent on a managed device connects to the Update Server to get updates only if the **Connect to the update server to update data** option is enabled in the properties of the *Find vulnerabilities and required updates* task and the **Windows Update search mode** option is set to **Active** in the settings of Network Agent policy.

- If you do not need Network Agent to initiate a connection to the Microsoft Windows update source and download updates when performing the *Vulnerability scan* task, you can set the **Windows Update search mode** option to **Passive**, while the **Connect to the update server to update data** option must remain enabled. This allows for you to save resources and use previously received Windows updates to scan for vulnerabilities. You can use the passive mode if you configure receiving Microsoft Windows updates in a different way. If receiving Microsoft Windows updates is not configured in another way, do not set the **Windows Update search mode** option to **Passive**, because in this case, information about updates will never be received.

- Irrespective of the **Connect to the update server to update data** option's status (enabled or disabled), if the **Windows Update search mode** option is set to **Disabled**, Open Single Management Platform does not request any information about updates.

If in the network of your organization the WSUS server is used (local WSUS server), and you want to use it together with Kaspersky Security Center Administration Server, you must first ensure that Kaspersky Security Center Administration Server is not used as a WSUS server, and then do the following:

1. On your managed devices running Windows, specify your local WSUS server for the Windows Update service. You can do it either manually in Windows Registry or by using a Windows group policy.

2. On your local WSUS server, specify the required classifications for updates and the applications on which the updates are to be distributed.

3. Synchronize your WSUS server with the upstream server (this can be either a Microsoft WSUS server or another local upstream WSUS server).

After the *Find vulnerabilities and required updates* task is run on the device, Network Agent uses the Windows Update service to initiate a request for Windows updates applicable to the device. The Windows Update service obtains the address of the external WSUS server from the registry and searches for updates directly from that external WSUS server. Then, the Windows Update service forwards this list to the Administration Server through Network Agent. Thus, Administration Server has an up-to-date base.

- **Search for third-party vulnerabilities and updates listed by Kaspersky**

If this option is enabled, Open Single Management Platform searches for vulnerabilities and required updates for third-party applications (applications made by software vendors other than Kaspersky and Microsoft) in Windows Registry and in the folders specified under **Specify paths for advanced search of applications in file system**. The full list of supported third-party applications is managed by Kaspersky.

If this option is disabled, Open Single Management Platform does not search for vulnerabilities and required updates for third-party applications. For example, you may want to disable this option if you have different tasks with different settings for Microsoft Windows updates and updates of third-party applications.

By default, this option is enabled.

- **Specify paths for advanced search of applications across the file system**

  The folders in which Open Single Management Platform searches for third-party applications that require vulnerability fix and update installation. You can use system variables.

  Specify the folders to which applications are installed. By default, the list contains system folders to which most of the applications are installed.

- **Enable advanced diagnostics**

  If this feature is enabled, Network Agent writes trace files even if writing trace files is disabled for Network Agent in Open Single Management Platform Remote Diagnostics Utility. Traces are written to two files in turn; the total size of both files is determined by the **Maximum size, in MB, of advanced diagnostics files** value. When both files are full, Network Agent starts writing to them again. The files with traces are stored in the %WINDIR%\Temp folder. These files are accessible in the remote diagnostics utility, you can download or delete them there.

  If this feature is disabled, Network Agent writes traces according to the settings in Open Single Management Platform Remote Diagnostics Utility. No additional traces are written.

  When creating a task, you do not have to enable advanced diagnostics. You may want to use this feature later if, for example, a task run fails on some of the devices and you want to get additional information during another task run.

  By default, this option is disabled.

- **Maximum size, in MB, of advanced diagnostics files**

  The default value is 100 MB, and available values are between 1 MB and 2048 MB. You may be asked to change the default value by Kaspersky Technical Support specialists when information in the advanced diagnostics files sent by you is not enough to troubleshoot the problem.

Recommendations on the task schedule

When scheduling the *Find vulnerabilities and required updates* task, make sure that two options—**Run missed tasks** and **Use automatically randomized delay for task starts**—are enabled.

By default, the *Find vulnerabilities and required updates* task is set to start manually. If the organization's workplace rules provide for shutting down all devices at this time, the *Find vulnerabilities and required updates* task will run after the devices are turned on again, that is, in the morning of the next day. Such activity may be undesirable because a vulnerability scan may increase the load on CPUs and disk subsystems. You must set up the most convenient schedule for the task based on the workplace rules adopted in the organization.

About the license

A *license* is a time-limited right to use Open Single Management Platform, granted under the terms of the signed License Contract (End User License Agreement).

The scope of services and validity period depend on the license under which the application is used.

The following license types are provided:

- *Trial*

  A free license intended for trying out the application. A trial license usually has a short term.

  When a trial license expires, all Open Single Management Platform features become disabled. To continue using the application, you need to purchase a commercial license.

  You can use the application under a trial license for only one trial period.

- *Commercial*

  A paid license.

  When a commercial license expires, key features of the application become disabled. To continue using Open Single Management Platform, you must renew your commercial license. After a commercial license expires, you cannot continue using the application and must remove it from your device.

  We recommend renewing your license before it expires, to ensure uninterrupted protection against all security threats.

OpenAPI Reference Guide

This OpenAPI reference guide is designed to assist in the following tasks:

- Automation and customization. You can automate tasks that you might not want to handle manually. For example, as an administrator, you can use OpenAPI to create and run scripts that will facilitate developing the structure of administration groups and keep that structure up-to-date.

- Custom development. Using OpenAPI, you can develop a client application.

You can use the search field in the right part of the screen to locate the information you need in the OpenAPI reference guide.

OPENAPI REFERENCE GUIDE

Samples of scripts

The OpenAPI reference guide contains samples of the Python scripts listed in the table below. The samples show how you can call OpenAPI methods and automatically accomplish various tasks for protecting your network, for instance, create a "primary/secondary" hierarchy, run tasks in Open Single Management Platform, or assign distribution points. You can run the samples as is or create your own scripts based on the samples.

To call the OpenAPI methods and run scripts,

Install the KlAkOAPI package from the KlAkOAPI.tar.gz archive on a device where Administration Server is installed.

The KlAkOAPI package includes a set of samples that perform typical operations. You can call the OpenAPI methods, run the samples and your own scripts only on devices where Administration Server and the KlAkOAPI package are installed.The KlAkOAPI.tar.gz archive that contains this package is located in the Open Single Management Platform installation folder.

Matching between user scenarios and samples of OpenAPI methods

Sample	Purpose of the sample	Scenario
Log KlAkParams	You can extract and process data by using the `KlAkParams` data structure. The sample shows how to work with this data structure. The sample output may be present in different ways. You can get the data to send an HTTP method or to use it in your code.	Monitoring and reporting
Create and delete a "primary/secondary" hierarchy	You can add a secondary Administration Server and establish a "primary/secondary" hierarchy. Alternately, you can disconnect the secondary Administration Server from the hierarchy.	Creating a hierarchy of Administration Servers, adding a secondary Administration Server, and deleting a hierarchy of Administration Servers
Download network list files via connection gateway to the specified host	You can connect to Network Agent on the needed device by using a connection gateway, and then download a file with the network list to your device.	Adjustment of distribution points and connection gateways
Install a license key stored in the primary Administration Server repository onto the secondary Administration Servers	You can connect to the primary Administration Server, download a required license key from it, and transmit this key to all the secondary Administration Servers included in a hierarchy.	Licensing of managed applications
Create a report of effective user rights	You can create different reports. For instance, you can generate the report of effective user rights by using this sample. This report describes the rights that a user has, depending on his or her group and role. You can download the report in the HTML, PDF, or Excel format.	Generating and viewing a report
Start the device task	You can connect to Network Agent on the needed device by using a connection gateway, and then run the necessary task.	Starting a task manually
Register distribution points for devices in a group	You can assign managed devices as distribution points (previously known as update agents).	Updating Kaspersky databases and applications
Enumerate all groups	You can perform various actions with administration groups. The sample shows how to do the following: • Get an identifier of the "Managed devices" root group • Move through the group hierarchy • Retrieve the full, expanded hierarchy of groups, along with their names and nesting	Configuring Administration Server
Enumerate tasks, query task statistics, and run a task	You can find out the following information: • Task progress history • Current task status • Number of tasks in different statuses You can also run a task. By default, the sample runs a task after it outputs statistics.	Starting a task manually

Sample	Purpose of the sample	Scenario
Create and run a task	You can create a task. Specify the following task parameters in the sample: • Type • Method of run • Name • Device group for which the task will be used By default, the sample creates a task with the "Show message" type. You can run this task for all managed devices of Administration Server. If necessary, you can specify your own task parameters.	Creating a task
Enumerate license keys	You can get a list of all the active license keys for Kaspersky applications installed on managed devices of Administration Server. The list contains detailed data about every license key, such as a name, type, or expiration date.	Licensing
Create a custom category	You can create the application category with the needed parameters.	Creating an application category with content added manually
Enumerate users by using SrvView	You can use the SrvView class to request detailed information from the Administration Server. For instance, you can get a list of users by using this sample.	Managing users and user roles

Applications interacting with Open Single Management Platform via OpenAPI

Some applications interact with Open Single Management Platform via OpenAPI. Such applications include, for example, Kaspersky Anti Targeted Attack Platform. This can also be a custom client application developed by you based on OpenAPI.

Applications interacting with Open Single Management Platform via OpenAPI connect to Administration Server. To find out whether the application that you use works by OpenAPI, see Help of this application.

Monitoring, reporting, and audit

This section describes the monitoring and reporting capabilities of Open Single Management Platform. These capabilities give you an overview of your infrastructure, protection statuses, and statistics.

After Open Single Management Platform deployment or during the operation, you can configure the monitoring and reporting features to best suit your needs.

Scenario: Monitoring and reporting

This section provides a scenario for configuring the monitoring and reporting feature in Open Single Management Platform.

Prerequisites

After you deploy Open Single Management Platform in an organization's network, you can start to monitor it and generate reports on its functioning.

Monitoring and reporting in an organization's network proceeds in stages:

1. **Configuring the switching of device statuses**

   Get acquainted with the settings for device statuses depending on specific conditions. By changing these settings, you can change the number of events with *Critical* or *Warning* importance levels. When configuring the switching of device statuses, be sure of the following:

   - New settings do not conflict with the information security policies of your organization.
   - You are able to react to important security events in your organization's network in a timely manner.

2. **Configuring notifications about events on client devices**

   How-to instructions:

   Configure notification (by email, by SMS, or by running an executable file) of events on client devices

3. **Performing recommended actions for Critical and Warning notifications**

   How-to instructions:

   Perform recommended actions for your organization's network

4. **Reviewing the security status of your organization's network**

   How-to instructions:

   - Review the Protection status widget
   - Generate and review the Report on protection status
   - Generate and review the Report on errors

5. **Locating client devices that are not protected**

How-to instructions:

- Review the New devices widget
- Generate and review the Report on protection deployment

6. **Checking protection of client devices**

   How-to instructions:

   - Generate and review reports from the Protection status and Threat statistics categories
   - Start and review the Critical event selection

7. **Evaluating and limiting the event load on the database**

   Information about events that occur during operation of managed applications is transferred from a client device and registered in the Administration Server database. To reduce the load on the Administration Server, evaluate and limit the maximum number of events that can be stored in the database.

   How-to instructions:

   - Limiting the maximum number of events

8. **Reviewing license information**

   How-to instructions:

   - Add the License key usage widget to the dashboard and review it
   - Generate and review the Report on usage of license keys

Results

Upon completion of the scenario, you are informed about protection of your organization's network and, thus, can plan actions for further protection.

About types of monitoring and reporting

Information on security events in an organization's network is stored in the Administration Server database. Based on the events, OSMP Console provides the following types of monitoring and reporting in your organization's network:

- Dashboard
- Reports
- Event selections
- Notifications

Dashboard

The dashboard allows you to monitor security trends on your organization's network by providing you with a graphical display of information.

Reports

The Reports feature allows you to get detailed numerical information about the security of your organization's network, save this information to a file, send it by email, and print it.

Event selections

Event selections provide an onscreen view of named sets of events that are selected from the Administration Server database. These sets of events are grouped according to the following categories:

- By importance level—**Critical events**, **Functional failures**, **Warnings**, and **Info events**
- By time—**Recent events**
- By type—**User requests** and **Audit events**

You can create and view user-defined event selections based on the settings available, in the OSMP Console interface, for configuration.

Notifications

Notifications alert you about events and help you to speed up your responses to these events by performing recommended actions or actions you consider as appropriate.

Triggering of rules in Smart Training mode

This section provides information about the detections performed by the Adaptive Anomaly Control rules in Kaspersky Endpoint Security for Windows on client devices.

The rules detect anomalous behavior on client devices and may block it. If the rules work in Smart Training mode, they detect anomalous behavior and send reports about every such occurrence to Administration Server. You can view the reports about detected anomalous behavior in **Operations** → **Repositories** → **Rule triggers in Smart Training state**. You can confirm detections as correct or add them as exclusions, so that this type of behavior is not considered anomalous anymore.

Information about detections is stored in the event log on the Administration Server (along with other events) and in the Adaptive Anomaly Control report.

For more information about Adaptive Anomaly Control, the rules, their modes and statuses, refer to Kaspersky Endpoint Security for Windows Help.

Viewing and confirming detections performed using Adaptive Anomaly Control rules

To view the list of detections performed by Adaptive Anomaly Control rules:

1. In the main menu, go to **Operations** → **Repositories** → **Rule triggers in Smart Training state**.

   The list displays the following information about detections performed using Adaptive Anomaly Control rules:

   - **Administration group**

     The name of the administration group where the device belongs.

   - **Virtual Administration Server**

     Virtual Administration Server that manages the device.

   - **Device name**

     The name of the client device where the rule was applied.

   - **Name**

     The name of the rule that was applied.

   - **Status**

     Excluding—If the Administrator processed this item and added it as an exclusion to the rules. This status remains till the next synchronization of the client device with the Administration Server; after the synchronization, the item disappears from the list.

     Confirming—If the Administrator processed this item and confirmed it. This status remains till the next synchronization of the client device with the Administration Server; after the synchronization, the item disappears from the list.

     Empty—If the Administrator did not process this item.

   - **Detections count**

     The number of detects within one heuristic rule, one process and one client device. This number is counted by Kaspersky Endpoint Security.

   - **User name**

     The name of the client device user who run the process that generated the detect.

- **Source process path**

  Path to the source process, i.e. to the process that performs the action (for more information, refer to the Kaspersky Endpoint Security help).

- **Source process hash**

  SHA256 hash of the source process file (for more information, refer to the Kaspersky Endpoint Security help).

- **Source object path**

  Path to the object that started the process (for more information, refer to the Kaspersky Endpoint Security help).

- **Source object hash**

  SHA256 hash of the source file (for more information, refer to the Kaspersky Endpoint Security help).

- **Target process path**

  Path to the target process (for more information, refer to the Kaspersky Endpoint Security help).

- **Target process hash**

  SHA256 hash of the target file (for more information, refer to the Kaspersky Endpoint Security help).

- **Target object path**

  Path to the target object (for more information, refer to the Kaspersky Endpoint Security help).

- **Target object hash**

  SHA256 hash of the target file (for more information, refer to the Kaspersky Endpoint Security help).

- **Processed**

  Date when the anomaly was detected.

To view the properties of a detection:

1. In the main menu, go to **Operations** → **Repositories** → **Rule triggers in Smart Training state**.

2. Do one of the following:

   - In the **Name** column, click the link with the name of the detection you want to view.

   - In the list of detections, select the check box next to the detection you want to view, and then click the **Properties** button.

   The properties window of the selected detection opens, displaying information about it.

You can confirm any detection from the list of detections of Adaptive Anomaly Control rules or from the properties window of a selected detection.

To confirm a detection:

- Select one or several detections in the list of detections, and then click the **Confirm** button.

- Open the properties window of a selected detection, and then click the **Confirm** button.

  The status of the detection is changed to **Confirming**. The detection will disappear from the list of detections after the next synchronization of the client device with the Administration Server.

  Your confirmation will contribute to the statistics used by the rules. For more information, refer to Kaspersky Endpoint Security for Windows Help.

Adding exclusions from the Adaptive Anomaly Control rules

The Add to Adaptive Anomaly Control exclusions wizard allows you to add exclusions from the Adaptive Anomaly Control rules for Kaspersky Endpoint Security.

To add exclusions from the Adaptive Anomaly Control rules by using the wizard:

1. Start the wizard in one of the following ways:

   - In the main menu, go to **Operations** → **Repositories** → **Rule triggers in Smart Training state**, select one or several detections, and then click the **Exclude** button.

     You can add up to 1000 exclusions at a time.

     Before adding a detection to exclusions, you can view the properties of the detection by clicking the detection name or the **Properties** button. In the detection properties window that opens, you can also click the **Exclude** button.

   - In the main menu, go to **Monitoring** → **Event selections**, click the link with the event selection you need, select the check box next to the detection you want to exclude, and then click the **Exclude from Adaptive Anomaly Control** button.

   The Add to Adaptive Anomaly Control exclusions wizard starts. Proceed through the wizard by using the **Next** button.

2. Select the policies and profiles to which you want to add exclusions.

   Inherited policies cannot be updated. If you do not have the rights to modify a policy, the policy will not be updated.

3. Click **Done** to close the wizard.

The status of the detection is changed to **Excluding**. The detection disappears from the list of detections after the next synchronization of the client device with the Administration Server. The exclusion from the Adaptive Anomaly Control rules is configured and applied.

Dashboard and widgets

This section contains information about the dashboard and the widgets that the dashboard provides. The section includes instructions on how to manage widgets and configure widget settings.

Using the dashboard

The dashboard allows you to monitor security trends on your organization's network by providing you with a graphical display of information.

The dashboard is available in the OSMP Console, in the **Monitoring** → **Dashboard** section.

The dashboard provides widgets that can be customized. You can choose a large number of different widgets, presented as pie charts or donut charts, tables, graphs, bar charts, and lists. The information displayed in the widgets is automatically updated, the update period is from one to two minutes. The interval between updates varies for different widgets. You can refresh data on a widget manually at any time by using the settings menu.

The dashboard includes the **Administration and protection** and **Detection and response** tabs, to which you can add widgets.

The Administration and protection tab

The **Administration and protection** tab can contain widgets that display information about all events stored in the database of Administration Server.

In the **Administration and protection** tab, the widgets of the following groups are available:

- **Protection status**
- **Deployment**
- **Updating**
- **Threat statistics**
- **Other**

The Detection and response tab

The **Detection and response** tab can contain widgets that display information about detected and registered alerts and incidents, and the response actions to them. You can view data only for those tenants to which you have access.

In the **Detection and response** tab, the widgets of the following groups are available:

- **Events**
- **Active lists**
- **Context tables**
- **Alerts**
- **Assets**
- **Incidents**
- **Playbooks**
- **Event sources**
- **Users**

Administration and protection widgets

When configuring the **Administration and protection** tab of the dashboard, you can add widgets, hide widgets, change the size or appearance of widgets, move widgets, and change their settings.

Some widgets have text information with links. You can view detailed information by clicking the link.

The following widget groups and widgets are available on the **Administration and protection** tab of the dashboard:

- **Protection status**

  The group includes the following widgets:

  - **History of software vulnerabilities**
  - **Number of vulnerable devices**
  - **Distribution of devices by severity level of vulnerabilities**
  - **Status of selected device**
  - **Protection status**

- **Deployment**

  This group includes the **New devices** widget.

- **Updating**

This group includes the following widgets:

- Statistics about Windows Update updates
- Distribution of anti-virus databases
- Active alerts
- Statistics of update installation results by update category
- Statistics of update installation statuses by update category
- Statistics of update installation statuses

- Threat statistics

  This group includes the following widgets:

  - Detection of threats by a specified application component distributed by disinfection result
  - Detection of threats by application components
  - Prohibited applications
  - Types of network attacks
  - Types of detected viruses and disinfection results
  - Quarantine history
  - History of detection of probably infected objects
  - History of network attacks
  - History of threat activity sorted by application type
  - Threat activity
  - Users of the 10 most heavily infected devices
  - Most heavily infected devices
  - Virtual Administration Servers infected most frequently
  - Most frequent threats
  - Windows domains infected most frequently
  - Groups infected most frequently
  - Alerts

- Other

This group includes the following widgets:

- **License key usage**
- **Notifications by selected severity level**
- **Top 10 most frequent events in database**
- **Current status of selected Administration Server task**
- **Task history**

Adding widgets to the dashboard

To add widgets to the dashboard:

1. In the main menu, go to **Monitoring** → **Dashboard**.
2. Click the **Add or restore widget** button.
3. In the list of available widgets, select the widgets that you want to add to the dashboard.

   Widgets are grouped by category. To view the list of widgets included in a category, click the chevron icon (>) next to the category name.
4. Click the **Add** button.

The selected widgets are added at the end of the dashboard.

You can now edit the representation and parameters of the added widgets.

Hiding a widget from the dashboard

To hide a displayed widget from the dashboard:

1. In the main menu, go to **Monitoring** → **Dashboard**.
2. Click the settings icon (⚙) next to the widget that you want to hide.
3. Select **Hide widget**.
4. In the **Warning** window that opens, click **OK**.

The selected widget is hidden. Later, you can add this widget to the dashboard again.

Moving a widget on the dashboard

To move a widget on the dashboard:

1. In the main menu, go to **Monitoring** → **Dashboard**.
2. Click the settings icon (⚙) next to the widget that you want to move.
3. Select **Move**.
4. Click the place to which you want to move the widget. You can select only another widget.

   The places of the selected widgets are swapped.

Changing the widget size or appearance

For widgets that display a graph, you can change its representation—a bar chart or a line chart. For some widgets, you can change their size: compact, medium, or maximum.

To change the widget representation:

1. In the main menu, go to **Monitoring** → **Dashboard**.
2. Click the settings icon (⚙) next to the widget that you want to edit.
3. Do one of the following:
   - To display the widget as a bar chart, select **Chart type: Bars**.
   - To display the widget as a line chart, select **Chart type: Lines**.
   - To change the area occupied by the widget, select one of the values:
     - **Compact**
     - **Compact (bar only)**
     - **Medium (donut chart)**
     - **Medium (bar chart)**
     - **Maximum**

   The representation of the selected widget is changed.

Changing widget settings

To change settings of a widget:

1. In the main menu, go to **Monitoring** → **Dashboard**.
2. Click the settings icon (⚙) next to the widget that you want to change.
3. Select **Show settings**.
4. In the widget settings window that opens, change the widget settings as required.
5. Click **Save** to save the changes.

   The settings of the selected widget are changed.

The set of settings depends on the specific widget. Below are some of the common settings:

- **Widget scope** (the set of objects for which the widget displays information)—for example, an administration group or device selection.
- **Select task** (the task for which the widget displays information).
- **Time interval** (the time interval during which the information is displayed in the widget)—between the two specified dates; from the specified date to the current day; or from the current day minus the specified number of days to the current day.
- **Set to Critical if the following settings are specified** and **Set to Warning if the following settings are specified** (the rules that determine the color of a traffic light).

After you change the widget settings, you can refresh data on the widget manually.

To refresh data on a widget:

1. In the main menu, go to **Monitoring** → **Dashboard**.
2. Click the settings icon (⚙) next to the widget that you want to move.
3. Select **Refresh**.

   The data on the widget is refreshed.

Detection and response widgets

On the **Detection and response** tab, you can add, configure, and delete widgets.

A selection of widgets used in the **Detection and response** tab is called a *layout*. All widgets must be placed in layouts. Kaspersky Next XDR Expert allows you to create, edit, and delete layouts. Preconfigured layouts are also available. You can edit widget settings in the preconfigured layouts as necessary. By default, the **Alerts Overview layout** is selected on the **Detection and response** tab.

The widget displays data for the period selected in the widget or layout settings only for the tenants that are selected in the widget or layout settings.

By clicking the link with the name of the widget about events, alerts, incidents, or active lists, you can go to the corresponding section of the Kaspersky Next XDR Expert interface. Note that this option is not available for some widgets.

The following widget groups and widgets are available on the **Detection and response** tab of the dashboard:

- **Events**. Widget for creating analytics based on events.
- **Active lists**. Widget for creating analytics based on active lists of correlators.
- Alerts. Group for analytics related to alerts. Includes information about alerts and incidents that is provided by Kaspersky Next XDR Expert.

  The group includes the following widgets:

  - **Active alerts**. Number of alerts that have not been closed.
  - **Active alerts by tenant**. Number of unclosed alerts for each tenant.
  - **Alerts by tenant**. Number of alerts of all statuses for each tenant.
  - **Unassigned alerts**. Number of alerts that have no assignee.
  - **Alerts by status**. Number of alerts that have the *New*, *Opened*, *Assigned*, or *Escalated* status. The grouping is by status.
  - **Latest alerts**. Table with information about the last 10 unclosed alerts belonging to the tenants selected in the layout.
  - **Alerts distribution**. Number of alerts created during the period configured for the widget.
  - **Alerts by users**. Number of alerts with the *Assigned* status. The grouping is by account name.
  - **Alerts by groups**. Number of alerts assigned to each group and to analysts within each group. Alerts assigned to an analyst are counted in every group where that analyst is a member.
  - **Alerts by severity**. Number of unclosed alerts grouped by their severity.
  - **Alerts by rule**. Number of unclosed alerts grouped by correlation rule.
  - **Alert statuses by period**. Number of alerts by status over the selected period.
  - **Alerts by technology**. Names of protection technologies that generated the alert.
  - **Alerts by hash**. Number of alerts grouped by the hashes of scanned objects. Supported hash types: MD5 and SHA256.
  - **Alerts by detect source**. Number of alerts grouped by the detection source.
  - **Alerts by file name**. Number of alerts grouped by scanned file name.

- **Assets**. Group for analytics related to assets from processed events. This group includes the following widgets:
  - **Affected assets in alerts**. Table with the names of assets and related tenants, and the number of unclosed alerts that are associated with these assets. The moving from the widget to the section with the asset list is not available.
  - **Affected asset categories**. Categories of assets linked to unclosed alerts.
  - **Number of assets**. Number of assets that were added to Kaspersky Next XDR Expert.
  - **Assets in incidents by tenant**. Number of assets associated with unclosed incidents. The grouping is by tenant.
  - **Assets in alerts by tenant**. Number of assets associated with unclosed alerts, grouped by tenant.
- Incidents. Group for analytics related to incidents.

  The group includes the following widgets:
  - **Active incidents**. Number of incidents that have not been closed.
  - **Unassigned incidents**. Number of incidents that have the *Opened* status.
  - **Incidents distribution**. Number of incidents created during the period configured for the widget.
  - **Incidents by status**. Number of incidents grouped by status.
  - **Incidents by type**. Number of incidents in any status grouped by type.
  - **Active incidents by tenant**. Number of unclosed incidents grouped by tenant available to the user account.
  - **All incidents**. Number of incidents of all statuses.
  - **All incidents by tenant**. Number of incidents of all statuses, grouped by tenant.
  - **Affected assets categories in incidents**. Asset categories associated with unclosed incidents.
  - **Latest incidents**. Table with information about the last 10 unclosed incidents belonging to the tenants selected in the layout.
  - **Incidents by user**. Number of incidents with the *Assigned* status. The grouping is by user account name.
  - **Incidents by groups**. Number of incidents assigned to each group and to analysts within each group. Incidents assigned to an analyst are counted in every group where that analyst is a member.
  - **Incidents by severity**. Number of unclosed incidents grouped by their severity.
  - **Affected assets in incidents**. Number of assets associated with unclosed incidents. The moving from the widget to the section with the asset list is not available.
  - **Affected users in incidents**. Users associated with incidents. The moving from the widget to the section with the user list is not available.

- **Event sources**. Group for analytics related to sources of events. The group includes the following widgets:
  - **Top event sources by alerts number**. Number of unclosed alerts grouped by event source.
  - **Top event sources by convention rate**. Number of events associated with unclosed alerts. The grouping is by event source.

    In some cases, the number of alerts generated by sources may be inaccurate. To obtain accurate statistics, it is recommended to specify the Device Product event field as unique in the correlation rule, and enable storage of all base events in a correlation event. However, correlation rules with these settings consume more resources.

- **Users**. Group for analytics related to users from processed events. The group includes the following widgets:
  - **Affected users in alerts**. Number of accounts related to unclosed alerts. The moving from the widget to the section with the user list is not available.
  - **Number of AD users**. Number of Active Directory accounts received via LDAP during the period configured for the widget.

    In the events table, in the event details area, in the alert window, and in the widgets, the names of assets, accounts, and services are displayed instead of the IDs as the values of the SourceAssetID, DestinationAssetID, DeviceAssetID, SourceAccountID, DestinationAccountID, and ServiceID fields. When exporting events to a file, the IDs are saved, but columns with names are added to the file. The IDs are also displayed when you point the mouse over the names of assets, accounts, or services. Searching for fields with IDs is only possible using IDs.

- **Playbooks**. Group for analytics related to playbooks.

  To view widgets in this group, you must have one of the following XDR roles: Main administrator, Tenant Administrator, SOC Administrator, SOC Manager, Tier 1 analyst, Tier 2 analyst, Approver, Auditor.

  The group includes the following widgets:

  - **Statistics MTTR**. Changes of the time to first response to alerts and incidents for the specified period of time (by default for 30 days). The widget displays a column chart.

    The following configuration parameters of the **Statistics MTTR** widget are available:

    - **MTTR type**:
      - **Mean**. Changes of the mean time to first response to alerts and incidents.
      - **Minimum**. Changes of the minimum time to first response to alerts and incidents.
      - **Maximum**. Changes of the maximum time to first response to alerts and incidents.

- **Response mode**:
  - **Manual**. Changes of the time only to manual first responses.
  - **Automatic**. Changes of the time only to automatic first responses.
  - **All**. Changes of the time to all first responses.
- **Scope**:
  - **Alerts**. Changes of the time to first response only to alerts.
  - **Incidents**. Changes of the time to first response only to incidents.
  - **All**. Changes of the time to first response to alerts and incidents.

- **Automatic and manual launches of playbooks**. The total number of automatic and manual launches of playbooks for a certain period. The widget displays a column chart.

  The **Launch type** parameter of the widget specifies whether to show only the number of automatic, only the number of manual, or the total number of playbook launches for a certain period.

  For the **Statistics MTTR** and **Automatic and manual launches of playbooks** widgets, you can also set the **Period segments length** parameter. This parameter specifies a time interval within which data will be grouped. You can group data for every hour, every 4 hours, or every 24 hours. On the column chart, the **Period segments length** parameter specifies the column width.

- **Coverage of alerts and incidents with playbooks**. Number of active alerts and incidents. You can select what components to display: incidents, alerts or all.

  The donut chart displays alerts/incidents in the following sectors:

  - Alerts/incidents for which a playbook in **Auto** operation mode was launched.
  - Alerts/incidents for which a playbook in **Training** operation mode was launched.
  - All other alerts/incidents.

- **Time saved by using playbooks**. Time saved by launching all the playbooks that have *Success* or *Warning* action status.

  The widget is not displayed by default.

You can view the full playbook list by clicking the name of any playbook widget.

- **IOA rules triggering Sandbox scan**. Top 10 IOA rules that most frequently cause Kaspersky Anti Targeted Attack Platform to submit files for scanning by the Sandbox component.
- **Top triggered IOA rules**. Top 10 IOA rules most frequently triggered in events detected by Targeted Attack Analyzer (TAA) technology.
- **Devices triggering IOA rules**. Top 10 devices on which IOA rules were triggered most frequently.

Basics of managing widgets

The principle of data display in the widget depends on the type of the graph. The following graph types are available in KUMA:

- Pie chart ( ).
- Counter ( ).
- Table ( ).
- Bar chart ( ).
- Date Histogram ( ).
- Line chart.
- Stacked bar chart.

Basics of general widget management

The name of the widget is displayed in the upper left corner of the widgets. By clicking the link with the name of the widget about events, alerts, incidents, or active lists, you can go to the corresponding section of OSMP Console.

A list of tenants for which data is displayed is located under the widget name.

In the upper right corner of the widget, the period for which data is displayed on the widget is indicated (for example, 30 days 30d). Keep in mind that the data displayed in the dashboard may lag behind real time because of caching. You can view the date and time of the last update by hovering over the period icon.

If the **Show data for previous period** setting is enabled for the widget, and the widget is displaying data for a relative period, the tooltip also displays the previous period. The previous period is calculated relative to the current period as start and end values of the current period minus the duration of the current period. For example, if data is updated daily and displayed for a month, but only the first 10 days of the month have passed, the previous period is taken to be the last 10 days of the previous month.

You can change the data display period for the widget by clicking the period icon and selecting an exact date or a relative period in the window that is displayed. If you want the widget to display data for the period selected for the layout, click the **Reset** button. Changing the displayed period on the layout also changes the period displayed in the widget.

> The time in the widget is displayed in the local time zone set in the browser.

The **CSV** button is located to the left of the period icon. You can download the data displayed on the widget in CSV format (UTF-8 encoding). The downloaded file name has the format <widget name>_<download date (YYYYMMDD)>_<download time (HHMMSS)>.CSV.

The widget displays data for the period selected in widget or layout settings only for the tenants that are selected in widget or layout settings.

If you have enabled the **Show total** toggle switch in the widget settings, the widget displays the **Total** value, but when you download the widget in CSV format, the **Total** value is not displayed. To display the **Total** value in the report, you can use Excel formulas or manually copy the total value from the widget.

Basics of managing Pie chart graphs

A pie chart with a legend is displayed below the list of tenants. The legend lists the names of the categories corresponding to the sectors of the diagram. By left-clicking a category in the legend, you can go to the section of OSMP Console with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

Under the period icon, the number of events, active lists, assets, alerts, or incidents grouped by the selected criteria for the data display period will be displayed.

Examples:

- In the **Alerts by status** widget, under the period icon, the number of alerts grouped by the **New**, **Open**, **Assigned**, or **Escalated** status is displayed.
  If you want to see the legend only for alerts with the **Opened** and **Assigned** status, you can clear the check boxes to the left of the **New** and **Escalated** statuses.
- In the **Events** widget, for which the SQL query `SELECT count(ID) AS `metric`, Name AS `value` FROM `events` GROUP BY Name ORDER BY `metric` DESC LIMIT 10` is specified, 10 events are displayed below the period icon, grouped by name and sorted in descending order.
  If you want to view events with specific names in the legend, you can clear the check boxes to the left of the names of events that you do not want to see in the legend.

Basics of managing Counter graphs

Graphs of this type display the sum total of selected data.

Example 1
The **Number of assets** widget displays the total number of assets added to KUMA.
Example 2
If you want to view trends to compare current data with data for a past period, you can create a widget that displays upward or downward trends. Trends can be configured in the following widgets:

- **Assets → Number of assets**.
- **Assets → Customized widget**.
- **Alerts → Active alerts**.
- **Alerts → Unassigned alerts**.
- **Incidents → Active incidents**.
- **Incidents → Unassigned incidents**.
- **Incidents → All incidents**.
- **Users -> Number of AD users**.
  To configure the display of trends, you need to enable the **Show data for previous period** toggle switch, and then select a trend on the Appearance tab. If the percentage value cannot be calculated, or if the values are equal, the (—) icon is displayed. If **Up is better** is selected, an uptrend is indicated by a green arrow on the chart, and a downtrend is indicated by a red arrow. If **Down is better** is selected, an uptrend is indicated by a red arrow on the chart, and a downtrend is indicated by a green arrow.
  You can click the value to change the display of data in the widget (for example, to display absolute values instead of percentages). This functionality is available when the widget is displayed on the dashboard and when TV mode is on.

Basics of managing Table graphs

Graphs of this type display data in a table format.

> Example:
> In the **Events** widget, for which the SQL query `SELECT TenantID , Timestamp , Name , DeviceProduct , DeviceVendor FROM `events` LIMIT 10` is specified, displays an event table with **TenantID**, **Timestamp**, **Name**, **DeviceProduct**, and **DeviceVendor** columns. The table contains 10 rows.

You can customize the display of data in the table using two-color (red and green) or three-color (red, yellow, and green) gradient fill. After selecting the type of gradient, you can adjust the direction of the fill by clicking the toggle direction icon (⇆) and set thresholds. Two values must be specified for a two-color gradient, and three for a three-color gradient. These values are displayed in the legend and determine the color of the marker added to the cell of the selected column. The column is specified in the **Field for gradient indicator** drop-down list.

Basics of managing Bar chart graphs

A bar chart with a legend is displayed below the list of tenants. The legend lists the names of the categories corresponding to the bars of the diagram. By left-clicking a category in the legend, you can go to the section of OSMP Console with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

> Example:
> In the **a Netflow top internal IPs** widget for which the SQL query `SELECT sum(BytesIn) AS metric, DestinationAddress AS value FROM `events` WHERE (DeviceProduct = 'netflow' OR DeviceProduct = 'sflow') AND (inSubnet(DestinationAddress, '10.0.0.0/8') OR inSubnet(DestinationAddress, '172.16.0.0/12') OR inSubnet(DestinationAddress, '192.168.0.0/16')) GROUP BY DestinationAddress ORDER BY metric DESC LIMIT 10` is specified, the x-axis of the chart corresponds to the total traffic in bytes, and the y-axis corresponds to destination port addresses. The data is grouped by destination address in descending order of total traffic.

You can also customize the display of bars using two-color (red and green) or three-color (red, yellow, and green) gradient fill. After selecting the type of gradient, you can adjust the direction of the fill by clicking the toggle direction icon (⇆) and set thresholds. Two values must be specified for a two-color gradient, and three for a three-color gradient. These values are displayed in the legend and determine the coloring of the bars.

Basics of managing Date Histogram graphs

A Date Histogram with a legend is displayed below the list of tenants. The legend lists the names of the categories corresponding to the bars of the diagram. By left-clicking a category in the legend, you can go to the section of OSMP Console with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

If, when creating a widget with a **Date Histogram** type, the **Bar chart** view is selected, you can customize the display of the bars using two-color (red and green) or three-color (red, yellow, and green) gradient fill. After selecting the type of gradient, you can adjust the direction of the fill by clicking the toggle direction icon (⇆) and set thresholds. Two values must be specified for a two-color gradient, and three for a three-color gradient. These values are displayed in the legend and determine the coloring of the bars.

Example:

```
SELECT count(ID) AS `metric`, Timestamp AS `value`
FROM `events`
GROUP BY Timestamp
ORDER BY `metric` DESC
LIMIT 250
```

The X axis of the chart shows the date when the event was created, and the Y axis shows the number of events. The bars of the chart are arranged from left to right in descending order of the number of events.

If an additional field with the `category` alias is used in the SQL query, the bar chart is displayed as a stacked bar chart. In this case, gradient fill is not applied.

Example:

```
SELECT count(ID) AS `metric`, Timestamp AS `category`, Type AS `value`
FROM `events`
GROUP BY Timestamp, Type
ORDER BY metric DESC
LIMIT 250
```

The X axis indicates the time intervals, and the Y axis indicates the number of events. Each segment of the bar corresponds to a value of the field marked as **Type**.

If, when creating a widget with a **Date Histogram** type, the **Line chart** view is selected, there are two options for displaying the chart: single line (if only the `metric` and `value` fields are specified in the query) and multiple lines (if an additional field with the `category` alias is used). This lets you compare values for different groups, depending on which field is used for grouping in the query. For example, correlation rules, users, or hosts can be used. No gradient fill is applied to such charts.

Example:

```
SELECT count(ID) AS `metric`, Timestamp AS `category`, CorrelationRuleName AS `value`
FROM `events`
WHERE Type = 3
GROUP BY Timestamp, CorrelationRuleName
ORDER BY metric DESC
LIMIT 250
```

The chart displays the intervals on the X-axis (the field specified as the `category`). The Y-axis displays the number of times the correlation rules were triggered (the field marked as `metric`). Each line on the chart corresponds to a unique value in the `value` field (in this case, the name of the correlation rule).

You can use the following time fields for grouping in a Date Histogram:

- `Timestamp`
- `DeviceCustomDate1`
- `DeviceCustomDate2`
- `EndTime`
- `FileCreateTime`
- `FileModificationTime`
- `FlexDate1`
- `OldFileCreateTime`
- `OldFileModificationTime`
- `DeviceReceiptTime`
- `StartTime`

> For convenience, we recommend using supported ClickHouse functions to round the values in the time field (**Timestamp**), for example, to 1 hour or 1 day.

For a **Date Histogram** widget with the **Line chart** view, you can also configure the display of the average value. To do this, on the **Appearance** tab, for the **Regression** setting, select **LOESS**.

Basics of managing Line chart graphs

A line chart with a legend is displayed below the list of tenants. The legend displays the names of the categories corresponding to the lines on the chart. By left-clicking a category in the legend, you can go to the section of OSMP Console with the corresponding data. The data in that section is sorted in accordance with the filters and/or search query specified in the widget.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

Example:

In the **Events** widget, for which the SQL query `SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` GROUP BY SourcePort ORDER BY `value` ASC LIMIT 250` is specified, the x-axis corresponds to the approximate port number, and the y-axis corresponds to the number of events. The data is grouped by port number in ascending order.

For a **Line chart** widget, you can also configure the display of the average value. To do this, on the **Appearance** tab, for the **Regression** setting, select **Linear**. If the chart displays multiple categories, the average is displayed for each category.

Basics of managing graphs of the Stacked bar chart type

A stacked bar chart with a legend is displayed below the list of tenants. The legend displays the names of the categories that make up parts of each bar. To the left of each category is a check box that lets you hide or show the category. The number of bars in the chart corresponds to the number of values in the selected grouping. The bars have captions. The color of the corresponding category in the bar is assigned automatically. When you hover over the zones of the bars, a tooltip is displayed with the value and a description of the value. You can left-click a category in the legend to go to the section of OSMP Console with the relevant data.

When going from the legend of the widget to the events details section, pay attention to the text of the legend item that you click to go to the events section. Any \ or ' characters used in the text are escaped in the text of the SQL query. However, the / character is not escaped.

The meaning of bar height depends on the **Format** setting:

- If the **Absolute values** format is configured, the height of the bars corresponds to the sum of the values of the measured figure.

- If the **Relative values, %** format is configured, all bars have the same height of 100%, and the relative heights of colored zones on the bars correspond to the ratios of the values.

If, when creating a custom widget based on the stacked bar chart, you selected the **Show data for previous period** option, and the standard `value`, `category`, `metric` aliases are used in the query, the chart displays previous-period data as separate bars. However, if instead of the standard `metric`, the query uses a custom metric calculation with non-standard aliases, the **Show data for previous period** is not taken into account when displaying the chart (see example queries below).

Examples:
When creating a custom widget of the **Stacked Bar chart** type based on an SQL query of an **Events** widget, the following rules apply:

- The field in the query that is marked as the `value` alias (required field) is used in the chart as the field by which the legend is created and the bars are divided into categories.
- The field in the query that is marked as the `category` alias (required field) is used in the chart as the field that defines the arrangement of bars along the X axis.
- The field in the query marked as the `metric` alias is used in the chart as the field that defines how bar areas are counted. In this case, the distribution is performed automatically.

However, you can manage the count by using standard aggregation functions (`sum`, `avg`, `min`, `max`, `count`) and your own arbitrarily named aliases as metrics instead of the standard `metric` alias (in this case, the display of data for the previous period is not supported).

Example 1:
For the **Events** widget, the following SQL query is specified with standard aliases, and the **Show data for previous period** option was selected when creating the widget:

```
SELECT count(ID) AS `metric`,

Type AS `value`,

TenantID AS `category`

FROM `events`

GROUP BY value, category

ORDER BY metric DESC
```

The X-axis stands for tenants (the field marked as the `category`), the Y-axis stands for the number of events of a certain type (the field marked as the value). Inside each bar, the quantity corresponding to a certain type is represented by a certain color. In the legend, the names of the event types by which the bars are divided are displayed as categories. Each category has an automatically assigned corresponding color on the bar. If you want to view only certain event types in the chart, you can clear or select the check boxes to the left of the corresponding event types in the legend.

Next to each bar, an additional bar is displayed with historical data, if such data was received in the query response.

Example 2:
For the **Events** widget, the following SQL query is marked with custom metrics marked as the `Base` and `Audit` aliases instead of the standard `metric` alias:

```
SELECT SUM(IF (Type = 1,1,0)) AS `Base`,

SUM(IF (Type = 4,1,0)) AS `Audit`,

TenantID as `category`

FROM `events`

GROUP BY category
```

The X-axis stands for tenants (the field marked as `category`), the Y-axis stands for the number of events of each type (custom metrics marked as the Base and Audit aliases). Inside each bar, the quantity corresponding to a certain metric is represented by a certain color. In the legend, the `Base` and `Audit` metrics by which the bars are divided into categories are displayed as category names. Each category has an automatically assigned corresponding color on the bar. If you want to view only event corresponding to a certain metric in the chart, you can clear or select the check boxes to the left of the relevant metric in the legend.

The additional bar with historical data is not displayed for a query with custom metrics, even if the Show data for previous period option was selected when creating the widget.

Example 3:
For the **Events** widget, the following SQL query is specified with standard aliases:

```
SELECT count(ID) AS `metric`,

TenantID as `value`,

Type as `category`

FROM events
```

```
GROUP BY value, category

ORDER BY metric DESC
```

In contrast to the similar query in example 1, in this case, the X-axis stands for the types of events (the field marked as the `category`), and the tenants (the field marked as the `value`) are represented by ranges of values in the bars (along the X axis) and the corresponding captions in the legend.

Example 4:
For the **Events** widget, the following SQL query is specified with standard aliases:

```
SELECT count(ID) AS `metric`,

CAST(fromUnixTimestamp64Milli(Timestamp) AS DATE) AS `category`,

Type as `value`

FROM `events`

GROUP BY category, value

ORDER BY category DESC
```

The chart displays the days of the month on the X axis (the field marked as the `category`). The Y axis displays the number of events of a particular type (the field marked as the `value`).

To create a similar chart with bars arranged by date and/or time, use a query with grouping and sorting by the following fields of the `timestamp` type:

- `Timestamp`
- `DeviceCustomDate1`
- `DeviceCustomDate2`
- `EndTime`
- `FileCreateTime`
- `FileModificationTime`
- `FlexDate1`
- `OldFileCreateTime`
- `OldFileModificationTime`
- `DeviceReceiptTime`
- `StartTime`

> We recommend using the **Date Histogram** to work with data that is arranged by date and/or time.

Basics of managing Speedometer charts

This type of chart displays the percentage of the number of objects of the selected category vs the total number of objects for a certain period. The **Speedometer chart** can be used for the following objects:

- Assets in a category that satisfy certain criteria vs the total number of assets in the same category or uncategorized assets
- Alerts that satisfy certain criteria vs the total number of alerts for a certain period

- Incidents that satisfy certain criteria vs the total number of incidents
- Events that satisfy certain criteria vs the total number of events for a certain period

The value on the chart is displayed as a percentage with a color matching the configured ranges of values. By default, the following ranges are specified: 0–30, 30–70, 70–100.

Example 1:
In the **Assets** customized widget, the user selects an asset category for comparing the number of assets in it with the total number of assets. The user can specify threshold values for the scale on the graph. Then, based on one or more conditions for the Y-axis, a calculation is performed with the existing asset database.

Example 2:
In the **Alerts** widget, you can select the **Speedometer chart** chart type for the **Active alerts** and **Unassigned alerts** categories. For alerts, a tenant and a period for analysis are selected. By default, these values match the values specified in the relevant dashboard. In the total number of alerts, you can also display data for the previous period.

The percentage value for the **Active incidents** and **Unassigned incidents** categories is calculated in the same way in the **Incidents** widget.

Example 3:
In the **Events** widget, for the **Speedometer chart** chart type, the user specifies a tenant, a period for analysis, and the event data storage. The SQL query is displayed as follows:

```
SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` GROUP BY SourcePort ORDER BY value ASC LIMIT
250
```

Special considerations for displaying data in widgets

Limitations for the displayed data

For improved readability, KUMA has limitations on the data displayed in widgets depending on its type:

- Pie chart displays a maximum of 20 slices.
- Bar chart displays a maximum of 40 bars.
- Table displays a maximum of 500 entries.
- Date histogram displays a maximum of 365 days.

Data that exceeds the specified limitations is displayed in the widget in the **Other** category.

You can download the full data used for building analytics in the widget in CSV format.

Summing up the data

The format of displaying the total sum of data on date histogram, bar chart and pie chart depends on the locale:

- English locale: decades (every three digits) are separated by commas, the decimal part is separated by a period.
- Russian locale: decades (every three digits) are separated by spaces, the decimal part is separated by a comma.

Creating a widget

You can create a widget in a dashboard layout while creating or editing the layout.

To create a widget:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Create a layout or switch to editing mode for the selected layout.
3. Click **Add widget**.
4. Select a widget type from the drop-down list.
   This opens the widget settings window.
5. Edit the widget settings.
6. If you want to see how the data will be displayed in the widget, click **Preview**.
7. Click **Add**.

The widget appears in the dashboard layout.

Editing a widget

To edit widget:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the edit button ( ).
   The **Customizing layout** window opens.
5. In the widget you want to edit, click the settings icon ( ).
6. Select **Edit**.
   This opens the widget settings window.
7. Edit the widget settings.
8. Click **Save** in the widget settings window.
9. Click **Save** in the **Customizing layout** window.

The widget is edited.

Deleting a widget

To delete a widget:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the edit button ( ).

   The **Customizing layout** window opens.
5. In the widget you want to delete, click the settings icon ( ).
6. Select **Delete**.
7. In the opened confirmation window, click **OK**.
8. Click the **Save** button.

The widget is deleted.

Widget settings

This section describes the settings of all widgets available in KUMA.

Events widget

You can use the **Events widget** to get analytics based on SQL queries.

When creating this widget, you must specify the settings described in the tables below.

General tab

The following table describes the settings on the **General** tab.

Description of parameters

Setting	Description
Graph	Graph type. The following graph types are available: • **Pie chart** • **Bar chart** • **Counter** • **Line chart** • **Table** • **Date Histogram** • **Stacked Bar chart** • **Speedometer chart**
Format	Data display format: **Absolute values** or **Relative values, %**. The setting is available for a **Stacked Bar chart**. If you select the **Absolute values** format, the heights of the bars correspond to the sum of the values of the measured indicator. If you select the Relative values, % format, all bars have the same height of 100%, and the relative heights of colored zones on the bars correspond to the ratios of indicator values. By default, **Absolute values** is selected.

Setting	Description
Timeline view	Time data display format: **Bar chart** or **Line chart**. This setting is available for charts of the **Date Histogram** type. When you select the **Bar chart** format, the data is displayed as vertical bars by default. If the SQL query contains an additional field with the category alias, the chart becomes a stacked bar chart, that is, each bar is divided into stripes in accordance with the values of this field. When you select the **Line chart** format, the data is displayed as one or more (depending on the query) polylines drawn through points on the time axis. The **Bar chart** format is selected by default.
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Time period	Period for which data is displayed in the widget. The default is **As layout**, meaning that data is displayed for the period selected for the layout. You can also specify a period for the widget in one of the following ways: • Select the exact start and end date and time of the period in the calendar and click **Apply**. • Select a period relative to the present time in the **Relative period** list. • Specify a value manually: select an exact date and time or a relative period, or a combination of both. For details, see the *Configuring a period* subsection below.
Show data for previous period	Enable the display of data for two periods at the same time: for the current period and for the previous period. When using a **Stacked Bar chart**, the **Show data for previous period** setting is taken into account if the query contains standard aliases: `value`, `category`, `metric`. However, if instead of the standard metric, the query uses a custom metric calculation with non-standard aliases, the **Show data for previous period** is not taken into account.
Defined variables	Local or global variables that you can use to filter widget data. Click **Add**. If you are adding a local variable, enter an arbitrary name for the variable, then enter the value of the variable. If you are adding a global variable, in the list, select the name of the variable that you specified in the dashboard properties.
Storage	Storage that is searched for events. The list displays the available spaces. You can select only one storage, but you can select one or more spaces. The values in the **Storage** field are independent of the selected tenants in the **Tenant** field. The field displays storages and spaces, like in the **Events** section. When a new space is created in the storage, this new space is not selected by default in the widget settings. You must manually select the new space. If the user has access to all spaces listed in the widget, the widget can display the event information for tenants specified in the space set in the user's permissions. If the user does not have access rights to one or more spaces of the storage, the widget cannot display information; the user cannot edit the widget, but can duplicate the widget using the **Duplicate** button. Duplication does not depend on access rights to spaces. If a template is duplicated in widgets that have spaces specified that are not accessible to the user, the value in the **Storage** field is reset. Such widgets display an error: Access denied (Operation returns no results because of allowed and selected event spaces). To save the template, you need to specify spaces in widgets. In widgets that have spaces that are accessible to the user, the value of the **Storage** field is not reset and is saved when the template is duplicated. When the user downloads the template, the data of inaccessible files are hidden in the downloaded file. If the user's email address is included in the list of recipients of the scheduled report, the user gets the full version of the report, regardless of which spaces are accessible.

Setting	Description

SQL query field	Query for filtering and searching for events manually. You can create a query in Builder by clicking **Use sql builder**.
	You can use extended event schema fields to create a chart of the **Table** type.
	Example
	Prerequisite:
	Events are coming from a FWSaaS system.
	10.08.2023 12:12:12 src=10.10.10.10 ReqUrls=[https://google.com, https:/ya.ru]
	10.08.2023 12:12:23 src=10.10.10.20 ReqUrls=[https://google.com, https:/yndex.ru]
	10.08.2023 12:12:33 src=10.10.10.30 ReqUrls=[https://google.com, https:/ya.ru, https://mail.ru]
	The KUMA General administrator has added an extended event schema field in KUMA in the **Settings** → **Other** → **Advanced event schema fields** section. At the KUMA normalizer, field mapping is configured for the **ReqUrls** and **SA.urls** fields.
	In KUMA, normalized events look as follows:
	DeviceReceiptTime=10.08.2023 12:12:12 SourceAddress=10.10.10.10 SA.urls=[https://google.com, https:/ya.ru]
	DeviceReceiptTime=10.08.2023 12:12:23 SourceAddress=10.10.10.20 SA.urls=[https://google.com, https:/yandex.ru]
	DeviceReceiptTime=10.08.2023 12:12:30 SourceAddress=10.10.10.30 SA.urls=[https://google.com, https:/ya.ru, https://mail.ru]
	You want to build a "Top URLs for 30 minutes" dashboard. Here is an example of a request in the **Events** section and in widgets:
	SELECT res.1 AS url, res.2 AS count FROM (SELECT arrayJoin(groupArray((url, count))) AS res FROM (SELECT url, count() AS count FROM (SELECT SA.urls AS urls FROM `events`) ARRAY JOIN urls AS url GROUP BY url) GROUP BY url)
	Result:
	url count
	google.com 3
	ya.ru 2
	mail.ru 1
	yandex.ru 1
	
	KUMA supports the following functions that allow using arrays in queries:
	• The `groupArray` function packs all values of the x argument into an array and is used in aggregate functions when grouping data.
	• The `arrayJoin` function expands the array into separate strings as a result of the query.
	You can use variables use variables in the SQL query. You can specify variables only when creating the query manually. You cannot use variables in the SQL builder.
	Example with variables
	Example SQL query with a local variable:
	`SELECT count(ID) AS 'metric', Name AS 'value', TenantID AS 'category' FROM 'events' WHERE filter(Name = $Name) GROUP BY Name, TenantID ORDER BY metric DESC LIMIT 250`
	This query uses the local variable `$Name`. Value of the **Name event** field are substituted into this variable. You can apply the variable by clicking the funnel icon ( ) on the widget. This opens a window; in that window, you can specify a value for the variable. The widget filters the selection by the selected value of the **Name** field.
	Note that if in the SQL query you specify a variable for a field that takes a numeric value, such a value must be converted to a string. You can do this using the `toString` function. Otherwise, the SQL query returns an error. Example:
	`SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` WHERE (filter(toString(SourcePort) like $SourcePort)) GROUP BY SourcePort ORDER BY value ASC LIMIT 250`
	In this query, the **SourcePort event** field has a numeric value, therefore it is converted to a string: `toString(SourcePort)`.
	
	For detailed information on creating an SQL query in the query constructor, see below.
	The following limitations apply:
	• The `metric` and `value` aliases in SQL queries cannot be edited for any type of event analytics widget, except tables.
	• Aliases in widgets with charts of the **Table** type can contain Latin and Cyrillic characters, as well as spaces. When using spaces or Cyrillic, the alias must be enclosed in quotation marks: `"An alias with a space"`, `` `Another alias` ``.
	• `ARRAY JOIN SQL` queries are not supported.
	• When displaying data for the previous period, sorting by the `count(ID)` parameter may not work correctly. We recommend sorting by the metric parameter. For example, `SELECT count(ID) AS "metric", Name AS "value" FROM `events` GROUP BY Name ORDER BY metric ASC LIMIT 250`.
	• In widgets with charts of the **Counter** type, you must specify the method of data processing for the values of the `SELECT` function: `count`, `max`, `min`, `avg`, `sum`.
	• In widgets with charts of the **Table** type, the gradient is drawn based on the values of a number or time field (for example, `Timestamp`) specified in the SQL query. This field can be selected from the **Field for gradient indicator** drop-down list (if the query contains multiple number or time fields, any one of these can be selected). After this, the cells of the selected column are highlighted with gradient fill markers. If a field is not present in the query, the **Appearance** tab does not display the **Gradient** and **Field for gradient indicator** fields.
	• In Stacked Bar chart widgets, `Timestamp` values are not displayed on the x-axis when using a standard query. If you want to display time along the X axis, we recommend using a **Line chart**. If you need to use a **Stacked Bar chart**, the `Timestamp` value must be converted to string: `toString(Timestamp) AS `category`` .

Configuring a period

To configure the data display period, do one of the following:

- If necessary, change the date and time in the **Time period** setting in one of the following ways:

  - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

    You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

  - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

    The period is calculated relative to the current time.

  - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

    You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

    In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: + (only in the **Range to** field), -, / (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the period **now-5d** to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

    > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

KUMA stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the data display period, data will be displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If you want to take your time zone into account when selecting a relative data display period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the **Range from** and **Range to** (if a value other than now is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

> If you need results up to 23:59:59:999 UTC of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000 UTC) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000 UTC) of the previous day to 00:00:00:000 UTC of the current day. You can view the date and time of the last data update and the exact period for which the data is displayed by hovering over the period icon in the widget.

If the Show data for previous period setting is enabled for the widget, and the widget is displaying data for a relative period, the tooltip also displays the previous period. The previous period is calculated relative to the current period as start and end values of the current period minus the duration of the current period. For example, if data is updated daily and displayed for a month, but only the first 10 days of the month have passed, the previous period is taken to be the last 10 days of the previous month.

- If you want the widget to display data for the period selected for the layout, click the **Reset** button. Changing the displayed period on the layout also changes the period displayed in the widget.

How to create a query in Builder

To create a query in Builder:

1. Specify the values of the following parameters:

   a. **SELECT**—event fields that should be returned. The number of available fields depends on the selected graph type.

      - In the drop-down list on the left, select the event fields for which you want to display data in the widget.
      - The middle field displays what the selected field is used for in the widget: **metric** or **value**.

        If you selected the **Table** graph type, in the middle fields, you must specify column names using ANSII-ASCII characters.

      - In the drop-down list on the right, you can select an operation to be performed on the data:
        - **count**—event count. This operation is available only for the **ID** event field. Used by default for line charts, pie charts, bar charts, and counters. This is the only option for date histogram.
        - **max** is the maximum value of the **event** field from the event selection.
        - **min** is the minimum value of the **event** field from the event selection.
        - **avg** is the average value of the **event** field from the event selection.
        - **sum** is the sum of event field values from the event selection.

   b. **SOURCE** is the type of the data source. Only the **events** value is available for selection.

c. **WHERE**—conditions for filtering events.

- In the drop-down list on the left, select the event field that you want to use for filtering.
- Select the necessary operator from the middle drop-down list. The available operators depend on the type of value of the selected event field.
- In the drop-down list on the right, enter the value of the condition. Depending on the selected type of field, you may have to manually enter the value, select it from the drop-down list, or select it on the calendar.

You can add search conditions by clicking **Add condition** or remove search conditions by clicking ×.

You can also add groups of conditions by clicking **Add group**. By default, groups of conditions are added with the **AND** operator, but you can change the it if necessary. Available values: **AND**, **OR**, **NOT**. Group conditions are deleted using the **Delete group** button.

d. **GROUP BY**—event fields or aliases to be used for grouping the returned data. This parameter is not available for Counter graph type.

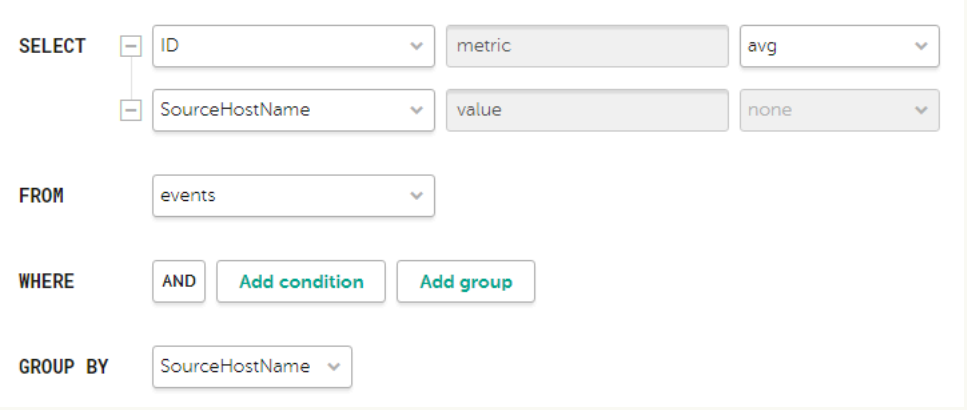
e. **ORDER BY**—columns used as the basis for sorting the returned data. This parameter is not available for the **Date Histogram** and **Counter** graph types.

- In the drop-down list to the left, select the value that will be used for sorting.
- Select the sort order from the drop-down list on the right: **ASC** for ascending, **DESC** for descending.
- For **Table** type graphs, you can add sorting conditions by clicking **Add column**.

f. **LIMIT** is the maximum number of data points for the widget. This parameter is not available for the **Date Histogram** and **Counter** graph types.

2. Click **Apply**.

Example of search conditions in the query builder



Parameters of search conditions for displaying the average number of bytes per host in the widget

Axes tab

The following table describes the settings on the **Axis** tab.

> The tab is displayed if on the **General** tab in the **Graph** field you have selected one of the following values: **Bar chart**, **Line chart**, **Date Histogram**.

Description of parameters

Setting	Description
Point size	Point size on the graph. This field is displayed for the **Line chart** graph type.
Line-width	Width of the line on the graph. This field is displayed for the **Line chart** graph type.
Minimum and maximum values for the Y-axis	Scale of the Y axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
Minimum and maximum values for the X-axis	Scale of the X axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
Decimals	Number of decimals to which the displayed value must be rounded off.

Appearance tab

The following table describes the settings on the **Appearance** tab.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default. Null values are displayed only when the **Show data for previous period** setting is enabled and there is no data for the current period, but there was some data for the previous period.
Color	The color used for displaying the information: • **default** for your browser's default font color • **green** • **red** • **blue** • **yellow** This setting is available for graphs such as **Bar chart**, **Counter**, **Line chart**, **Date Histogram**.
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) This setting is available for charts such as **Bar chart**, **Date Histogram** (with the **Bar chart** type), **Table**. You can adjust the direction of the fill by clicking the toggle direction icon (⇆); you can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold.
Field for gradient indicator	The column in which the gradient fill indicators are displayed. This setting is available for charts of the **Table** type.
Horizontal	Makes the histogram horizontal instead of vertical. When this option is enabled, when a widget displays a large amount of data, horizontal scrolling is not available and all available information is fit into the fixed size of the widget. If there is a lot of data to display, it is recommended to increase the widget size.

Setting	Description
Show total	Shows sums total of the values.
Period segments length	Duration of the segments into which you want to divide the period. Available for graphs of the **Date Histogram** type.
Scale	Scale for displaying data. Available for a **Stacked Bar** chart. The following values are possible: • **Linear** • **Square** • **Logarithm** The default is **Linear**.
Regression	An option for indicating the average value on the chart. The following values are possible: • **Not set** • **Linear** • **LOESS** (curve with locally estimated scatterplot smoothing). Available for the **Date Histogram** and **Line chart** types.

Actions tab

This tab is not displayed if you have selected one of the following values on the **General** tab, in the Graph field: **Counter**, **Speedometer chart**, **Table**.

Settings on this tab let you configure left-click actions on various elements of widgets, for example, on a chart segment or the text of a legend item. You can select the following actions:

- Go to events.
- Filter by variable.

  The settings are described in the scenario for filtering data in multiple widgets.
- Drill down to another dashboard.

  The settings are described in the scenario for drilling down to another dashboard.

Active lists widget

You can use the **Active lists** widget to get analytics based on SQL queries.

When creating this widget, you must specify the settings described in the tables below.

General tab

The following table lists the settings that must be specified on the **Common** tab.

Description of parameters

Setting	Description
Graph	Graph type. The following graph types are available: • **Bar chart** • **Pie chart** • **Counter** • **Table**
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Correlator	The name of the correlator that contains the active list for which you want to receive data.

Setting	Description
Active list	The name of the active list for which you want to receive data. The same active list can be used by different correlators. However, a separate entity of the active list is created for each correlator. Therefore, the contents of the active lists used by different correlators differ even if the active lists have the same names and IDs.
SQL query field	This field lets you manually enter a query for filtering and searching active list data. The query structure is similar to that used in event search. When creating a query based on active lists, you must consider the following: • For the FROM function, you must specify the `records` value. • If you want to receive data for fields whose names contain spaces and Cyrillic characters, you must also enclose such names in quotes in the query: • In the SELECT function, enclose aliases in double quotes or backticks: "alias", `another alias`. • In the ORDER BY function, enclose aliases in backticks: `another alias`. • Event field values are enclosed in straight single quotes: WHERE DeviceProduct = 'Microsoft'. • Names of event fields do not need to be enclosed in quotes. • If the name of an active list field begins or ends with spaces, these spaces are not displayed by the widget. The field name must not contain spaces only. • If the values of the active list fields contain trailing or leading spaces, it is recommended to use the LIKE '%field value%' function to search by them. • In your query, you can use service fields: _key (the field with the keys of active list records) and _count (the number of times this record has been added to the active list), as well as custom fields. • The "metric" and "value" aliases in SQL queries cannot be edited for any type of active lists analytics widget, except tables. • If a date and time conversion function is used in an SQL query (for example, fromUnixTimestamp64Milli) and the field being processed does not contain a date and time, an error will be displayed in the widget. To avoid this, use functions that can handle a null value. Example: SELECT _key, fromUnixTimestamp64Milli(toInt64OrNull(DateTime)) as Date FROM `records` LIMIT 250. • Large values for the LIMIT function may lead to browser errors. • If you select Counter as the graph type, you must specify the method of data processing for the values of the SELECT function: `count`, `max`, `min`, `avg`, `sum`. • In widgets with a chart of the **Table** type, the gradient is drawn based on the values of the field specified in the SQL query (for example, `_count`, `value`, `temp`, or `name`). This field can be selected from the **Field for gradient indicator** drop-down list (if the query contains multiple number or time fields, any one of these can be selected). After this, the cells of the selected column are highlighted with gradient fill markers. If a field is not present in the query, the **Appearance** tab does not display the **Gradient** and **Field for gradient indicator** fields. Special considerations apply when using aliases in SQL functions and SELECT, you can use double quotes and backticks: ", `.  If you selected Counter as the graph type, aliases can contain Latin and Cyrillic characters, as well as spaces. When using spaces or Cyrillic, the alias must be enclosed in quotation marks: "An alias with a space", `Another alias`.  When displaying data for the previous period, sorting by the count(ID) parameter may not work correctly. It is recommended to sort by the metric parameter. For example, SELECT count(ID) AS "metric", Name AS "value" FROM `events` GROUP BY Name ORDER BY metric ASC LIMIT 250.

You can get the names of the tenants in the widget instead of their IDs.

If you want the names of tenants to be displayed in active list widgets instead of tenant IDs, in correlation rules of the correlator, configure the function for populating the active list with information about the corresponding tenant.

To configure the function:

1. Export the list of tenants.
2. Create a dictionary of the **Table** type and import the previously obtained list of tenants into the dictionary.

3. Add a local variable with the **dict** function for mapping the tenant name to tenant ID to the correlation rule.

   Example:

   - Variable: `TenantName`
   - Value: `dict ('<Name of the previously created dictionary with tenants>', TenantID)`

4. Add an action with active lists to the correlation rule. T

   his action will write the value of the previously created variable in the key-value format to the active list using the **Set** function. As the key, specify the field of the active list (for example, `Tenant`), and in the value field, reference the previously created variable (for example, `$TenantName`).

   When this rule triggers, the name of the tenant mapped by the **dict** function to the ID from the tenant dictionary is placed in the active list. When creating widgets for active lists, you can get the name of the tenant by referring to the name of the field of the active list (in the example above, `Tenant`).

The method described above can be applied to other event fields with IDs.

Examples of SQL queries for receiving analytics based on active lists:

- `SELECT * FROM `records` WHERE "Event source" = 'Johannesburg' LIMIT 250`
  This query returns the key of the active list where the field name is "Event source" and the value of this field is "Johannesburg".
- `SELECT count(_key) AS metric, Status AS value FROM `records` GROUP BY value ORDER BY metric DESC LIMIT 250`
  Query for a pie chart, which returns the number of keys in the active list ('count' aggregation over the '_key' field) and all variants of the Status custom field. The widget displays a pie chart with the total number of records in the active list, divided proportionally by the number of possible values for the **Status** field.
- `SELECT Name, Status, _count AS Number FROM `records` WHERE Description ILIKE '%ftp%' ORDER BY Name DESC LIMIT 250`
  Query for a table, which returns the values of the **Name** and **Status** custom fields, as well as the service field '_count' for those records of the active list in which the value of the Description custom field matches ILIKE '%ftp%'. The widget displays a table with the **Status**, **Name**, and **Number** columns.

Axes tab

The following table lists the settings that must be specified on the **Axis** tab.

This tab is displayed if on the **Common** tab, in the **Graph** field, you have selected **Bar chart**.

Description of parameters

Settings	Description
Y-min and Y-max	Scale of the Y axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
X-min and X-max	Scale of the X axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.

Appearance tab

The following table lists the settings that must be specified on the **Appearance** tab.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Color	The color used for displaying the information: • **default** for your browser's default font color • **green** • **red** • **blue** • **yellow**
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) The setting is available for Bar charts and Tables. You can adjust the direction of the fill by clicking the toggle direction icon (⇆); you can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold.
Field for gradient indicator	The column in which the gradient fill indicators are displayed. This setting is available for charts of the **Table** type.
Horizontal	Makes the histogram horizontal instead of vertical. When this setting is enabled, all available information is fitted into the configured widget size. If the amount of data is great, you can increase the size of the widget to display it optimally.
Show total	Shows sums total of the values.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default. Null values are displayed only when the **Show data for previous period** setting is enabled and there is no data for the current period, but there was some data for the previous period.

Context tables widget

You can use the **Context tables** widget to get analytics based on SQL queries.

When creating this type of widget, you must set values for the following settings:

Tab :

The following table lists the settings that must be specified on the **Common** tab.

Description of parameters

Setting	Description
Graph	The following graph types are available: • **Bar chart** • **Pie chart** • **Counter** • **Table**
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Correlator	Name of the correlator that contains the context table for which you want to receive information.
Context table	Name of the context table for which you want to receive information. The same context table can be used in multiple correlators. However, a separate entity of the context table is created for each correlator. Therefore, the contents of the context tables used by different correlators are different even if the context tables have the same name and ID.

Setting	Description
SQL query field	This field lets you manually enter a query for filtering and searching context table data. By default, for each widget type, the field contains a query that obtains the context table schema and the key by key fields. The query structure is similar to that used in event search. When creating a query based on context tables, you must consider the following: • For the `FROM` function, you must specify the `` `records` `` value. • You can get data only for the fields specified in the context table schema. • You can use supported features of ClickHouse. • If you want to receive data for fields whose names contain spaces and Cyrillic characters, you must also enclose such names in quotes in the query: • In the `SELECT` function, enclose aliases in double quotes or backticks: "`<alias>`", `` `<another alias>` ``. • In the `ORDER BY` function, enclose aliases in backticks: `` `<another alias>` ``. • Event field values are enclosed in straight quotes: `WHERE DeviceProduct = 'Microsoft'`. • Names of event fields do not need to be enclosed in quotes. • If the name of an active list field begins or ends with spaces, these spaces are not displayed by the widget. The field name must not contain spaces only. • If the values of the active list fields contain trailing or leading spaces, it is recommended to use the `LIKE '%< field value >%'` function to search by them. • You can use the `_count` service field (how many times this record has been added to the context table), as well as custom fields. • The `metric` and `value` aliases in SQL queries cannot be edited for any type of active lists analytics widget, except tables. • If a date and time conversion function is used in an SQL query (for example, `fromUnixTimestamp64Milli`) and the field being processed does not contain a date and time, an error will be displayed in the widget. To avoid this, use functions that can handle a null value. Example: `SELECT _key, fromUnixTimestamp64Milli(toInt64OrNull(DateTime)) as Date FROM` `` `records` `` `LIMIT 250`. • Large values for the `LIMIT` function may lead to browser errors. • If you select **Counter** as the graph type, you must specify the method of data processing for the values of the `SELECT` function: `count`, `max`, `min`, `avg`, `sum`. • In widgets with a chart of the **Table** type, the gradient is available only for fields of the **INT** and **FLOAT** types and for the **_count** field. You can select the field in the **Field for gradient indicator** drop-down list. After this, the cells of the selected column are highlighted with gradient fill markers. Special considerations when using aliases in SQL functions and `SELECT` statements: you may use double quotes and backquotes: ",`. When using spaces or Cyrillic characters, the alias must be enclosed in double quotes: `<Alias with a space>`", values must be enclosed in straight single quotes: '`<Value with a space>`' When displaying data for the previous period, sorting by the count(ID) parameter may not work correctly. It is recommended to sort by the `metric` parameter. For example, `SELECT count(ID) AS "metric", Name AS "value" FROM` `` `events` `` `GROUP BY Name ORDER BY metric ASC LIMIT 250`.

You can get the names of the tenants in the widget instead of their IDs.

If you want the names of tenants to be displayed in active list widgets instead of tenant IDs, in correlation rules of the correlator, configure the function for populating the active list with information about the corresponding tenant.

To configure the function:

1. Export the list of tenants.
2. Create a dictionary of the **Table** type and import the previously obtained list of tenants into the dictionary.
3. Add a local variable with the **dict** function for mapping the tenant name to tenant ID to the correlation rule.

   Example:

   - Variable: `TenantName`
   - Value: `dict ('<Name of the previously created dictionary with tenants>', TenantID)`

4. Add an action with active lists to the correlation rule. T

   his action will write the value of the previously created variable in the key-value format to the active list using the **Set** function. As the key, specify the field of the active list (for example, `Tenant`), and in the value field, reference the previously created variable (for example, `$TenantName`).

   When this rule triggers, the name of the tenant mapped by the **dict** function to the ID from the tenant dictionary is placed in the active list. When creating widgets for active lists, you can get the name of the tenant by referring to the name of the field of the active list (in the example above, `Tenant`).

The method described above can be applied to other event fields with IDs.

Sample SQL queries for receiving analytics based on active lists:

- `SELECT * FROM `records` WHERE "Event source" = 'Johannesburg' LIMIT 250`
  This query returns the key of the active list where the field name is "`Event source`" and the value of this field is "`Johannesburg`".
- `SELECT count(_key) AS metric, Status AS value FROM `records` GROUP BY value ORDER BY metric DESC LIMIT 250`
  Query for a pie chart, which returns the number of keys in the active list ('`count`' aggregation over the '**_key**' field) and all variants of the Status custom field. The widget displays a pie chart with the total number of records in the active list, divided proportionally by the number of possible values for the **Status** field.
- `SELECT Name, Status, _count AS Number FROM `records` WHERE Description ILIKE '%ftp%' ORDER BY Name DESC LIMIT 250`
  Query for a table, which returns the values of the **Name** and **Status** custom fields, as well as the service field '**_count**' for those records of the active list in which the value of the Description custom field matches `ILIKE '%ftp%'`. The widget displays a table with the **Status**, **Name**, and **Number** columns.

Tab :

The following table lists the settings that must be specified on the **Axis** tab.

This tab is displayed if on the **Common** tab, in the **Graph** field, you have selected **Bar chart**.

Description of parameters

Setting	Description
Y axis and **Min - Max**	Scale of the Y axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
X axis and **Min - Max**	Scale of the X axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.

Tab

The following table lists the settings that must be specified on the **Appearance** tab.

Description of parameters

Settings	Description
Name	Name of the widget.
Description	Description of the widget.

Settings	Description
Color	The color used for displaying the information: • **default** for your browser's default font color • **green** • **red** • **blue** • **yellow**
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) The setting is available for **Bar charts** and **Tables**. You can adjust the direction of the fill by clicking the toggle direction icon (); y⇆ ou can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold.
Field for gradient indicator	The column in which the gradient fill indicators are displayed. This setting is available for charts of the **Table** type.
Horizontal	Makes the histogram horizontal instead of vertical. When this setting is enabled, all available information is fitted into the configured widget size. If the amount of data is great, you can increase the size of the widget to display it optimally.
Show total	Shows sums total of the values.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default. Null values are displayed only when the **Show data for previous period** setting is enabled and there is no data for the current period, but there was some data for the previous period.

Assets customized widget

You can use the **Assets** → **Customized widget** to get advanced asset analytics from processed events using the query builder. In the query, you need to specify the asset field and the corresponding condition or set of conditions by which you want the assets to be counted (Y-axis). You can also specify one or more additional conditions (categories) to be used for comparing the number of assets for each field.

When creating the custom widget for assets, you must specify the settings described in the tables below.

General tab

The following table describes the settings on the **General** tab.

Description of parameters

Setting	Description
Name	Name of the widget.
Graph	Graph type. The following graph types are available: • **Pie chart**. The distribution of assets by the specified categories is displayed for all selected tenants. • **Counter**. Assets are counted based on a single condition. The chart displays the sum of assets that match the condition across all selected tenants. • **Table**. Assets are counted by the specified categories across all selected tenants. Each category in the chart has a corresponding row. The category name is displayed in the **value** column. The number of assets matching the condition is displayed in the **metric** column. • **Stacked Bar chart** Assets are counted by the specified categories across all selected tenants, and you can group assets by tenants (in this case, individual tenants are arranged along the X axis). Each category has its own zone on the bars of the chart, represented by its own color. Category names are displayed in the legend. You can use check boxes to the left of category names to hide or show the corresponding areas on the bars. • **Speedometer chart** Displays the ratio of the number of assets that match the specified criteria to the total number of assets. The values on the graph are indicated as percentages.
Format	The setting is available for a **Stacked Bar chart**. Data display format: **Absolute values** or **Relative values, %**. If you select the **Absolute values** format, the heights of the bars correspond to the sum of the values of the measured indicator. If you select the **Relative values, %** format, all bars have the same height of 100%, and the relative heights of colored zones on the bars correspond to the ratios of indicator values. By default, **Absolute values** is selected.
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Asset category	The asset category for which the widget is displaying data.
Search in uncategorized assets	This setting lets you display assets that do not have a category. This check box is cleared by default.
Selecting Graphic Axes group of settings	
Y-axis	Required setting. Asset field and the condition or set of conditions specified for this field, that define how assets are to be counted.
Y axis category	Category for the selected field. Not used for a graph of the **Counter** type. This setting is optional for Y-axis fields whose values are enumerations (can be selected from a finite list of values). For all other fields, this parameter is required.
Group by tenant	This setting is available for charts of the **Stacked Bar chart** type. Enables additional grouping of assets by tenant. If the check box is selected, assets on the chart are broken up along the X-axis into bars corresponding to individual tenants. If the check box is cleared, all assets are displayed on the same bar. This check box is cleared by default.

Appearance tab.

The following table describes the settings on the **Appearance** tab.

Description of parameters

Setting	Description
Description	Description of the widget.
Show total	This setting is available for charts of the **Pie chart** type. Enables the display of totals in the chart, in addition to the selected categories. If this check box is enabled, the sum of the values of all specified categories is displayed in the center of the pie chart and in the legend in a separate **Total** column. The toggle switch is turned off by default.
Color	This setting is available for charts of the **Counter** type. The color used for displaying the information: • **default** for your browser's default font color • **green** • **red** • **blue** • **yellow**
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) This setting is available for charts of the **Table** type. You can adjust the direction of the fill by clicking the toggle direction icon (⇆); you can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold.
Field for gradient indicator	The column in which the gradient fill indicators are displayed. This column is determined by the **Metric** field selected in the drop-down list. This setting is available for charts of the **Table** type.
Horizontal	Makes the histogram horizontal instead of vertical. When this option is enabled, when a widget displays a large amount of data, horizontal scrolling is not available and all available information is fit into the fixed size of the widget. If there is a lot of data to display, it is recommended to increase the widget size.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default. Null values are displayed only when the **Show data for previous period** setting is enabled and there is no data for the current period, but there was some data for the previous period.
Show percentages in legend	Displays percentage values in the legend. The toggle switch is turned on by default.
Decimals	Number of decimals to which the displayed value must be rounded off.
Scale	Available for a **Stacked Bar chart**. Scale for displaying data. The following values are possible: • **Linear** • **Square** • **Logarithm** The default is **Linear**.
Threshold	Available for a **Speedometer chart**. By default, the following ranges are specified: 0–30, 30–70, 70–100.

Other widgets

This section describes the settings of all widgets except the **Events**, Active lists, Context tables, and **Custom Asset** widget.

The set of parameters available for a widget depends on the type of graph that is displayed on the widget. The following graph types are available in KUMA:

- Pie chart ( ).
- Counter ( ).
- Table ( ).
- Bar chart ( ).
- Date Histogram ( ).
- Line chart.
- Stacked bar chart.

Settings for pie charts

The following table below lists the settings of a **Pie chart**.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Time period	Period for which data is displayed in the widget. The default is **As layout**, meaning that data is displayed for the period selected for the layout. You can also specify a period for the widget in one of the following ways: • Select the exact start and end date and time of the period in the calendar and click **Apply**. • Select a period relative to the present time in the **Relative period** list. • Specify a value manually: select an exact date and time or a relative period, or a combination of both. For details, see the *Configuring a period* subsection below.
Show total	Shows sums total of the values.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default.
Decimals	Number of decimals to which the displayed value must be rounded off.

Settings for counters

The following table below lists the settings of a **Counter**.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.

Setting	Description
Time period	Period for which data is displayed in the widget. The default is **As layout**, meaning that data is displayed for the period selected for the layout. You can also specify a period for the widget in one of the following ways: • Select the exact start and end date and time of the period in the calendar and click **Apply**. • Select a period relative to the present time in the **Relative period** list. • Specify a value manually: select an exact date and time or a relative period, or a combination of both. For details, see the *Configuring a period* subsection below.

Settings for tables

The following table below lists the settings of a **Table**.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Tenant	The tenant for which data is displayed in the widget. You can select multiple tenants. By default, data is displayed for tenants selected in layout settings.
Time period	Period for which data is displayed in the widget. The default is **As layout**, meaning that data is displayed for the period selected for the layout. You can also specify a period for the widget in one of the following ways: • Select the exact start and end date and time of the period in the calendar and click **Apply**. • Select a period relative to the present time in the **Relative period** list. • Specify a value manually: select an exact date and time or a relative period, or a combination of both. For details, see the *Configuring a period* subsection below.
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) You can adjust the direction of the fill by clicking the toggle direction icon (⇆); you can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold.
Field for gradient indicator	The column in which the gradient fill indicators are displayed. This column is determined by the field selected in the drop-down list. The type of the field depends on the type of the widget: • For **Affected assets**, it is the **Frequency** field. • For **Affected assets in incidents**, **Real-time protection extended status of KSC assets**, **Affected users in incidents**, it is the **Count** field.
Decimals	Number of decimals to which the displayed value must be rounded off.

Settings for Bar charts and Date Histograms

The table below lists the settings for the **Bar chart** and **Date Histogram** type graphs located on the **Axis** tab.

Description of parameters

Setting	Description
Y axis and **Min - Max**	Scale of the Y axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
X axis and **Min - Max**	Scale of the X axis. Negative values can be displayed on chart axes. This is due to the scaling of charts on the widget and can be fixed by setting zero as the minimum chart values instead of **Auto**.
Decimals	Number of decimals to which the displayed value must be rounded off.

The table below lists the settings for the **Bar chart**, **Stacked Bar chart**, and **Date Histogram** type graphs located on the **Appearance** tab.

Description of parameters

Setting	Description
Name	Name of the widget.
Description	Description of the widget.
Time period	Period for which data is displayed in the widget. The default is **As layout**, meaning that data is displayed for the period selected for the layout. You can also specify a period for the widget in one of the following ways: • Select the exact start and end date and time of the period in the calendar and click **Apply**. • Select a period relative to the present time in the **Relative period** list. • Specify a value manually: select an exact date and time or a relative period, or a combination of both. For details, see the *Configuring a period* subsection below.
Format	Data display format: **Absolute values** or **Relative values, %**. The setting is available for a **Stacked Bar chart**. If you select the **Absolute values** format, the heights of the bars correspond to the sum of the values of the measured indicator. If you select the **Relative values, %** format, all bars have the same height of 100%, and the relative heights of colored zones on the bars correspond to the ratios of indicator values. By default, **Absolute values** is selected.
Timeline view	Time data display format: **Bar chart** or **Line chart**. This setting is available for charts of the **Date Histogram** type. When you select the **Bar chart** format, the data is displayed as vertical bars by default. If the SQL query contains an additional field with the category alias, the chart becomes a stacked bar chart, that is, each bar is divided into stripes in accordance with the values of this field. When you select the **Line chart** format, the data is displayed as one or more (depending on the query) polylines drawn through points on the time axis. The Bar chart format is selected by default.
Show data for previous period	Enable the display of data for two periods at the same time: the current period and the previous period.
Color	The color used for displaying the information: • **default** for your browser's default font color • **green** • **red** • **blue** • **yellow**
Gradient	Gradient fill of the chart: • Two colors (red and green) • Three colors (red, yellow, and green) You can adjust the direction of the fill by clicking the toggle direction icon (⇆); you can also set thresholds. If not set manually, gradient thresholds are calculated automatically from the data. If there is no data between the specified thresholds, the gradient is not applied. Instead, two colors are used: one for values below the lower threshold, and the other for values above the upper threshold. Gradient fill is not applied to stacked bar charts. In this case, colors are assigned automatically.
Horizontal	Makes the histogram horizontal instead of vertical. When this setting is enabled, all available information is fitted into the configured widget size. If the amount of data is great, you can increase the size of the widget to display it optimally.
Show total	Shows sums total of the values.
Show legend	Displays a legend for the analytics. The toggle switch is turned on by default.

Setting	Description
Show nulls in legend	Displays parameters with a null value in the legend for analytics. The toggle switch is turned on by default. Null values are displayed only when the **Show data for previous period** setting is enabled and there is no data for the current period, but there was some data for the previous period.
Period segments length	Duration of the segments into which you want to divide the period. Available for graphs of the **Date Histogram** type.

Settings for Speedometer charts

The table below lists the settings for a **Speedometer chart**.

The values on the graph are indicated as percentages.

Description of parameters

Setting	Description
General tab	
Name	Name of the widget.
Tenant	The tenant whose data is displayed in the widget. You can only select tenants that you selected earlier in the dashboard settings.
Asset category	Assets of the selected category in which values will be searched.
Search in uncategorized assets	When searching for assets for the chart, asset categories are disregarded.
Y-axis	The value of the parameter by which the search for assets and subsequent calculation of the speedometer reading will be performed. You can specify multiple parameter values to search for assets.
Appearance tab	
Description	Description of the chart that is displayed on the widget.
Threshold	To help you visually identify the severity of an event, the dial of the speedometer is divided into three areas: green, yellow, and red. By default, the following intervals are configured for these areas: • Green: 0–30 • Yellow: 30–70 • Red: 70–100 You can change the ranges as necessary.

Configuring a period

For graphs such as **Pie chart**, **Counter**, **Table**, **Bar chart**, **Stacked Bar chart**, **Date Histogram**, you can configure the period for displaying data in the widget using the **Period** setting. By default, the data display period of the widget is the same as the data display period of the dashboard.

To configure the data display period, do one of the following:

- If necessary, change the date and time in the **Time period** setting in one of the following ways:

  - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

    You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

  - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

    The period is calculated relative to the current time.

  - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

    You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

    In the relative period formulas, you can use the **now** parameter for the current date and time and the interval parameterization language: + (only in the **Range to** field), -, / (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the period **now-5d** to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

    > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

KUMA stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the data display period, data will be displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If you want to take your time zone into account when selecting a relative data display period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the **Range from** and **Range to** (if a value other than now is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

> If you need results up to 23:59:59:999 UTC of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000 UTC) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000 UTC) of the previous day to 00:00:00:000 UTC of the current day. You can view the date and time of the last data update and the exact period for which the data is displayed by hovering over the period icon in the widget.

If the Show data for previous period setting is enabled for the widget, and the widget is displaying data for a relative period, the tooltip also displays the previous period. The previous period is calculated relative to the current period as start and end values of the current period minus the duration of the current period. For example, if data is updated daily and displayed for a month, but only the first 10 days of the month have passed, the previous period is taken to be the last 10 days of the previous month.

If you want the widget to display data for the period selected for the layout, click the **Reset** button. Changing the displayed period on the layout also changes the period displayed in the widget.

Filtering widget data

KUMA allows you to filter data in widgets of the **Events** type by selecting filtering parameters in the dashboard. You can filter data in the following cases:

- Filtering data on a single widget

  For this type of filtering, you need to specify one or more local variables in the SQL query in widget properties. Variables allow you to modify this SQL query by substituting the data from the widget on the dashboard into the query. The widget filters the data based on the new values of the variables and refreshes the chart. You do not need to manually change the SQL query in the properties of the widget.

  You can apply local variables by clicking the funnel icon ( ) in the widget. This opens a window in which you can select or enter values for local variables.

- Filtering data in multiple widgets within the same dashboard

  For this type of filtering, you need to specify one or more global variables in dashboard properties. Then add global variables to the SQL queries of the widgets to which you want to apply these global variables. When you assign a new value to a global variable, all widgets that use this variable automatically rebuild charts based on the new value.

  Global variables are applied by clicking on a segment of the chart or on text from the legend of the chart. KUMA displays a menu with commands that apply the data category of the selected chart segment or plot legend item to a variable.

- Drilling down to another dashboard while passing filtering parameters

  Global variables can be used to pass values to another dashboard. When drilling down from one dashboard to another, KUMA substitutes the selected value of the variable and the selected time period, and the charts on the new dashboard are automatically rebuilt.

  Local variables cannot be used when drilling down to another dashboard.

> We do not recommend setting more than five variables within a dashboard. This can lead to unstable operation of the dashboard.

Filtering data in a single widget

KUMA allows you to filter data in widgets of the **Events** type by selecting filtering parameters in the dashboard.

For filtering data on a single widget, you need to specify one or more local variables in the SQL query in widget properties. Variables allow you to modify this SQL query by substituting the data from the widget on the dashboard into the query. The widget filters the data based on the new values of the variables and refreshes the chart. You do not need to manually change the SQL query in the properties of the widget.

> We do not recommend setting more than five variables within a dashboard. This can lead to unstable operation of the dashboard.

Configuring data filtering on a single widget

To configure local variables for data filtering within a widget:

1. In the **Dashboards** section, select a dashboard and click the **Edit** button to enter the dashboard editing mode.
2. Select an existing **Events** widget or click the **Add widget** button and then select the **Events** widget type.
3. In the widget properties window, select the **General** tab.
4. Under **Defined variables**, click **Add**.
5. Enter an arbitrary name for the variable, then specify the value of the variable. Local and global variable names must be unique within a dashboard.

   As the value of the variable, specify "`value`" or "`category`", that is, the field aliases used in the query. If you want to enter the value of this variable manually when filtering data in the widget, leave this field empty.

   For charts of the table type, you can specify the name of the event field as the value of the variable. This lets you filter the data in the table by the selected value of this field. The name of the event field is displayed in the header of the table column.
6. In the **SQL query** field, in the WHERE clause, add a function to handle the variables.

   The syntax for variables with a specific value is as follows:

   `filter(< event_field_name > = $ variable_name )`

   The syntax for variables with an arbitrary value to be entered manually is as follows:

   `filter(< event_field_name > like $ variable_name )`

   In this case, when typing manually, you can use the % metacharacter to search for partial text matches. For detailed information, see the ClickHouse documentation.

   > You cannot use the SQL builder to add variables to an SQL query.

   View example

   Example SQL query with a local variable:

   ```
   SELECT count(ID) AS 'metric', Name AS 'value', TenantID AS 'category' FROM 'events'
   WHERE filter(Name = $Name) GROUP BY Name, TenantID ORDER BY metric DESC LIMIT 250
   ```

   This query uses the local variable `$Name`. The value of the Name event field is substituted into this variable. You can apply the variable by clicking the funnel icon ( ) on the widget. This opens a window; in that window, you can specify a value for the variable. The widget filters the selection by the selected value of the Name field.

   Note that if in the SQL query you specify a variable for a field that takes a numeric value, such a value must be converted to a string. You can do this using the `toString` function. Otherwise, the SQL query returns an error. Example:

   ```
   SELECT count(ID) AS `metric`, SourcePort AS `value` FROM `events` WHERE
   (filter(toString(SourcePort) like $SourcePort)) GROUP BY SourcePort ORDER BY value
   ASC LIMIT 250
   ```

In this query, the SourcePort event field has a numeric value, therefore it is converted to a string: `toString(SourcePort)`.

7. Save your changes.

Applying data filtering to a single widget

To apply a local variable for data filtering within a widget:

1. In the **Dashboards** section, select the dashboard that contains the relevant widget. At least one local variable must be set in the properties of this widget.

2. On the widget, click the funnel icon ( ).

3. This opens a window; in that window, select local variables that you want to apply.

4. Save your changes.

   The widget rebuilds the chart taking into account the new value of the variable.

Filtering data in multiple widgets

KUMA allows you to filter data in widgets of the **Events** type by selecting filtering parameters in the dashboard.

To filter data in multiple widgets within one dashboard, you need to specify one or more global variables in the properties of the dashboard. Then add global variables to the SQL queries of the widgets to which you want to apply these global variables. When you assign a new value to a global variable, all widgets that use this variable automatically rebuild charts based on the new value.

Global variables are applied by clicking on a segment of the chart or on text from the legend of the chart. KUMA displays a menu with commands that apply the data category of the selected chart segment or plot legend item to a variable.

> We do not recommend setting more than five variables within a dashboard. This can lead to unstable operation of the dashboard.

Configuring data filtering for multiple widgets at the dashboard level

To configure global variables for filtering data at the dashboard level:

1. In the **Dashboards** section, select a dashboard and click the **Edit** button to enter the dashboard editing mode.

2. Click the **Settings** button.

3. This opens a window; in that window, in the **Variables** section, select the **Global variables** tab.

4. Click **Add**.

5. In the new row, enter the name of the variable. Local and global variable names must be unique within a dashboard.

   The names of all dashboard widgets to which you add global variables are listed in the **Used in widgets** column.

   The names of the global variables are displayed in the commands of the context menu when you click a segment of the chart or an item in the legend of the chart.

6. Save your changes.

7. Select an existing **Events** widget or click the **Add widget** button and then select the **Events** widget type. In this widget, you will be configuring actions on click.

8. In the widget properties window, select the **General** tab.

9. In the **Defined variables** section, add one or more global variables that you have added in the dashboard properties.

10. If necessary, in the SQL query field, add global variables to the WHERE clause using the filter() function. This also applies the configured actions to this widget.

11. Select the **Actions** tab.

    If you want to display a menu with a command to filter data by the selected segment category when a segment of the chart is clicked:

    a. In the **Chart segment** section, enable the **Filter by variable** option.

    b. In the **Category** list, select the variable into which you want to substitute the value of the `category` alias specified in the SQL query.

       The **Category** option is available only for the stacked bar chart type.

    c. In the **Value** list, select the variable into which you want to substitute the value of the `value` alias specified in the SQL query.

    If you want to display a menu with a command to filter data by the selected value when a legend item is clicked:

    a. In the **Legend** item section, enable the **Filter by variable** option.

       If necessary, click the **Copy from chart segment settings** button to automatically transfer the selected parameter values from the **Chart segment** group of settings to the **Legend item** group of settings.

    b. In the **Value** list, select the variables in which you want to be able to substitute legend item variables.

12. In the properties of other widgets to which you want to apply global variables:

    - On the **General** tab, in the **Defined variables** section, add one or more global variables.

    - In the SQL query field of the widget, add global variables to the WHERE clause using the filter() function.

Applying data filtering to multiple widgets at the dashboard level

To apply a global variable to filter data between widgets:

1. In the **Dashboards** section, select the dashboard to whose widgets you want to apply the global variable.

2. Find a widget that has actions on click configured. Click a segment of the graph or on the arrow icon (⌄) next to the legend item to open a menu.

3. In the menu, click **Apply category to variable <variable name>** or **Apply value to variable <variable name>**.

   The value of the selected category or legend item is assigned to the variable. Within the current dashboard, all widgets that use this variable rebuild their charts based on the new value of the variable. Such widgets are marked with a funnel icon with a number indicating the number of filters applied (1). A list of applied global variables is displayed in the upper part of the screen. To clear data filtering for a widget, click the funnel icon with the number (1). To clear data filtering for all widgets on the dashboard, click **Reset all** in the upper part of the dashboard.

Drilling down to a related dashboard

KUMA allows filtering data in widgets of the **Events** type in the current dashboard and drill down to another dashboard while keeping the configured filtering parameters and period.

This is achieved using global variables. The global variables must be configured in both dashboards, as well as in the properties of widgets: on the target dashboard, the variable is specified in the SQL query, and in the source dashboard, the variable is specified when managing the **Drill down to another dashboard** setting. When drilling down from one dashboard to another, KUMA automatically substitutes the selected value of the variable and the period, after which the charts in the target dashboard are rebuilt. Both dashboards must be in the same tenant.

Local variables cannot be used when drilling down to another dashboard.

Configuring the target dashboard

To configure the target dashboard:

1. In the KUMA web interface, in the **Dashboards** section, select a dashboard and create a global variable.

   The variable name may contain only English letters and numerals and may not contain spaces.

2. Save your changes.

3. On the dashboard, select an existing **Events** widget or click the **Add widget** button and then select the **Events** widget type.

4. On the **General** tab, specify the previously created variable:

   - Under **Defined variables**

   - In the **SQL query** field For the SQL query to take into account the value of the variable, add the `filter()` function to the `WHERE` clause.

     The syntax for variables with a specific value is as follows:

     ```
     filter(< event_field_name > = $ variable_name )
     ```

The syntax for variables with an arbitrary value entered manually is as follows:

`filter(< event_field_name > like $ variable_name )`

Query example

```
Query with a $ServiceName variable:
SELECT DestinationHostName AS Host, Name AS Status, Message AS Reason,
ServiceName, Timestamp AS Time
FROM events
WHERE DeviceEventClassID = 'KLSRV_HOST_STATUS_CRITICAL' OR DeviceEventClassID =
'KLSRV_HOST_STATUS_WARNING' AND filter(ServiceName = $ServiceName)
ORDER BY Time DESC
LIMIT 25
If a value is passed in the $ServiceName variable when drilling down to the target dashboard, the value
is automatically substituted into the query and the data is filtered. If the value of the variable is not
specified, the query is executed as if the variable was not present.
```

5. Save your changes.

Configuring the source dashboard

To configure the source dashboard:

1. In the KUMA web interface, in the **Dashboards** section, select a dashboard and create a global variable.

   The variable name must be the same as the name of the variable created in the target dashboard. The name may contain only English letters and numerals and may not contain spaces.

2. Save your changes.

3. On the dashboard, select an existing **Events** widget or click the **Add widget** button and then select the **Events** widget type.

4. On the **General** tab, in the **Defined variables** section, specify the previously created variable.

5. On the **Actions** tab, specify the following settings:

   a. In the **Chart segment** or **Legend item** section, enable **Drill down to another dashboard**.

   b. In the **Dashboards** drop-down list, select the target dashboard.

   c. In the **Value** drop-down list, select the variable whose value you want to pass when drilling down to another dashboard. This must be the same variable that you created earlier.

      The selected variable must be specified in the SQL query of the widget on the target dashboard, so that the data on this widget is automatically filtered after the drill-down.

6. Save your changes.

Drilling down from source dashboard to target dashboard

To drill down to the target dashboard:

1. In the KUMA web interface, under **Dashboards**, select the source dashboard.

2. Find the widget for which you have enabled the **Drill down to another dashboard** option. Click the segment or item in the legend of the chart for which this action is configured.

   This opens the target dashboard in a new tab. The period of the original widget is automatically applied to the entire dashboard, and the selected value from the **Value** drop-down list is passed as a variable to widgets of the **Event** type, in which this variable is used in the SQL query through the `filter()` function. The charts in the widgets on the other dashboards are rebuilt with the passed values taken into account.

   > If you change or delete a variable that is specified in the action on the target dashboard to which you are drilling down to, the drill-down is still performed, but the value of the variable is not passed.

Displaying tenant names in Active list type widgets

If you want the names of tenants to be displayed in **Active list** type widgets instead of tenant IDs, in correlation rules of the correlator, configure the function for populating the active list with information about the corresponding tenant.

The configuration process involves the following steps:

1. Export the list of tenants.

2. Create a dictionary of the **Table** type.

3. Import the list of tenants obtained at step 1 into the dictionary created at step 2 of these instructions.

4. Add a local variable with the **dict** function for mapping the tenant name to tenant ID to the correlation rule.

   Example:

   - Variable: `TenantName`

   - Value: `dict ('<Name of the previously created dictionary with tenants>', TenantID)`

5. Add a **Set** action to the correlation rule, which writes the value of the previously created variable to the active list in the <key>-<value> format. As the key, specify the field of the active list (for example, `Tenant`), and in the **Value** field, specify the variable (for example, `$TenantName`).

When this rule triggers, the name of the tenant mapped by the **dict** function to the ID in the tenant dictionary is placed in the active list. When creating widgets based on active lists, the widget displays the name of the tenant instead of the tenant ID.

Detection and response dashboard

In the **Dashboard** section, you can monitor the security status of your organization's network.

The dashboard is a set of widgets that display network security data analytics. You can view data only for those tenants to which you have access.

A selection of widgets used in the dashboard is called a *layout*. You can create layouts manually or use predefined layouts. You can edit widget settings in predefined layouts as necessary. By default, the dashboard displays the Alerts Overview predefined layout.

Only users with the Main administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, edit, or delete layouts. Users accounts with all roles can view layouts and set default layouts. If a layout is set as default, that layout is displayed for the account every time the user navigates to the **Dashboard** section. The selected default layout is saved for the current user account.

The information on the dashboard is updated in accordance with the schedule configured in layout settings. If necessary, you can force the update of the data.

For convenient presentation of information on the dashboard, you can enable TV mode. This mode lets you view the dashboard in full-screen mode in FullHD resolution. In TV mode, you can also configure a slide show display for the selected layouts.

Creating a dashboard layout

To create a layout:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.

2. Open the drop-down list in the top right corner of the window and select **Create layout**.

   The **New layout** window opens.

3. In the **Tenants** drop-down list, select the tenants that will own the created layout and whose data will be used to fill the widgets of the layout.

   The selection of tenants in this drop-down list does not matter if you want to create a universal layout (step 8).

4. In the **Time period** drop-down list, select the time period from which you require analytics:

   - **1 hour**
   - **1 day** (this value is selected by default)
   - **7 days**
   - **30 days**
   - **In period**—receive analytics for the custom time period. The time period is set using the calendar that is displayed when this option is selected.

     > The upper boundary of the period is not included in the time slice defined by it. In other words, to receive analytics for a 24-hour period, you should configure the period as Day 1, 00:00:00 – Day 2, 00:00:00 instead of Day 1, 00:00:00 – Day 1, 23:59:59.

5. In the **Refresh every** drop-down list, select how often data should be updated in layout widgets:

   - **1 minute**
   - **5 minutes**
   - **15 minutes**

- **1 hour** (this value is selected by default)
- **24 hours**

6. In the **Add widget** drop-down list, select the required widget and configure its settings.

   You can add multiple widgets to the layout.

   You can also drag widgets around the window and resize them using the diagonal button (⤡) that appears when you hover the mouse over a widget.

   You can edit or delete widgets added to the layout. To do this, click the settings icon (⚙) and select **Edit** to change their configuration or **Delete** to delete them from the layout.

   To add a widget:

   1. Click the Add widget drop-down list and select required widget.

      The window with widget parameters opens. You can see how the widget will look like by clicking the Preview button.

   2. Configure widget parameters and click the Add button.

   To add a widget:

   1. Click the **Add widget** drop-down list and select required widget.

      The window with widget parameters opens. You can see how the widget will look like by clicking the **Preview** button.

   2. Configure the widget parameters and click the **Add** button.

7. In the **Layout name** field, enter a unique name for this layout. Must contain 1 to 128 Unicode characters.

   A custom dashboard cannot have the same name as a predefined dashboard.

   An error is returned when trying to create a dashboard with the same name as a predefined dashboard or change the name of a custom dashboard to the name of a predefined dashboard in the Main tenant. In tenants other than Main, you can create or duplicate a custom dashboard with the same name as a predefined dashboard.

8. If necessary, click the settings icon (⚙) on the right of the layout name field, and then select the check box next to the **Universal** setting.

   The layout widgets display data from tenants that you select in the **Selected tenants** section in the menu on the left. This means that the data in the layout widgets will change based on your selected tenants without having to edit the layout settings. For universal layouts, tenants selected in the **Tenants** drop-down list are not taken into account.

   If the check box is cleared, layout widgets display data from the tenants that are selected in the **Tenants** drop-down list in the layout settings. If any of the tenants selected in the layout are not available to you, their data will not be displayed in the layout widgets.

   You cannot use the **Active Lists** widget in universal layouts.
   Universal layouts can only be created and edited by a user who has been assigned the Main administrator role. Such layouts can be viewed by all users.

9. Click **Save**.

The new layout is created and is displayed on the **Detection and response** tab of the dashboard.

Selecting a dashboard layout

To select a dashboard layout:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Expand the list in the upper right corner of the window.
3. Select the relevant layout.

The selected layout is displayed on the **Detection and response** tab of the dashboard.

Selecting a dashboard layout as the default

To set a dashboard layout as the default:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the star icon (☆).

The selected layout is displayed on the **Detection and response** tab of the dashboard by default.

Editing a dashboard layout

To edit a dashboard layout:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the edit icon (✎).

   The **Customizing layout** window opens.
5. Edit the dashboard layout. The settings that are available for editing are the same as the settings available when creating a layout.
6. Click **Save**.

The dashboard layout is edited and displayed on the **Detection and response** tab.

> If the layout is deleted or assigned to a different tenant while you are editing it, an error is displayed when you click **Save**. The layout is not saved. Refresh the Kaspersky Next XDR Expert interface page to see the list of available layouts in the drop-down list.

Deleting a dashboard layout

To delete layout:

1. In the main menu, go to **Monitoring** → **Dashboard**, and the select the **Detection and response** tab.
2. Expand the list in the upper right corner of the window.
3. Hover the mouse cursor over the relevant layout.
4. Click the delete icon ( ) and confirm this action.

The layout is deleted.

Enabling and disabling TV mode

> It is recommended to create a separate user with the minimum required set of right to display analytics in TV mode.

To enable TV mode:

1. In the KUMA Console, select the **Dashboard** section.
2. Click the button in the upper-right corner.
   The **Settings** window opens.
3. Move the **TV mode** toggle switch to the **Enabled** position.
4. To configure the slideshow display of the layouts, do the following:
   a. Move the **Slideshow** toggle switch to the **Enabled** position.
   b. In the **Timeout** field, indicate how many seconds to wait before switching layouts.
   c. In the **Queue** drop-down list, select the layouts to view. If no layout is selected, the slideshow mode displays all layouts available to the user one after another.
   d. If necessary, change the order in which the layouts are displayed by clicking the button to drag and drop them.
5. Click the **Save** button.

TV mode will be enabled. To return to working with the KUMA Console, disable TV mode.

To disable TV mode:

1. Open the KUMA Console and select the **Dashboard** section.
2. Click the button in the upper-right corner.
   The **Settings** window opens.

3. Move the **TV mode** toggle switch to the **Disabled** position.

4. Click the **Save** button.

TV mode will be disabled. The left part of the screen shows a pane containing sections of the KUMA Console.

When you make changes to the layouts selected for the slideshow, those changes will automatically be applied to the active slideshow sessions.

Preconfigured dashboard layouts

Kaspersky Next XDR Expert includes a set of predefined layouts that contain the following widgets:

- **Alerts Overview** layout (Alert overview):
  - Active alerts—number of alerts that have not been closed.
  - Unassigned alerts—number of alerts that have no assignee.
  - Latest alerts—table with information about the last 10 unclosed alerts belonging to the tenants selected in the layout.
  - Alerts distribution—number of alerts created during the period configured for the widget.
  - Alerts by priority—number of unclosed alerts grouped by their priority.
  - Alerts by users—number of alerts with the **In progress** status. The grouping is by account name.
  - Alerts by status—number of alerts grouped by their status.
  - Alerts by groups—number of alerts grouped by user groups.
  - Affected users in alerts—users associated with unclosed alerts.
  - Affected assets—table with information about assets, their importance level, and the number of associated unclosed alerts.The importance level of an asset is calculated based on the categories the asset belongs to, including all parent categories up to the root of the hierarchy.
  - Affected assets categories—chart displaying the number of asset categories specified in unclosed alerts and incidents. For each asset referenced in alerts or incidents, the importance level of an asset is calculated based on the categories the asset belongs to, including all parent categories up to the root of the hierarchy.
  - Top event source by alerts number—number of unclosed alerts grouped by the event source that generated the alerts.

    The widget displays up to 10 event sources.

  - Alerts by rule—number of unclosed alerts grouped by correlation rules.

Unclosed alerts are alerts that have one of the following statuses: **New**, **In progress**, **In incident**.

- **Incidents Overview** layout (Incidents overview):
  - Active incidents—number of incidents that are not in the **Done** status category.
  - Unassigned incidents—number of incidents that have no assignee.
  - Latest incidents—table with information about the last 10 unclosed incidents belonging to the tenants selected in the layout.
  - Incidents distribution—number of incidents created during the period configured for the widget.
  - Incidents by priority—number of incidents grouped by their priority.
  - Incidents by users—number of incidents with the **In progress** status. The grouping is by user account name.
  - Incidents by status—number of incidents grouped by their status.
  - Incidents by groups—number of incidents grouped by their user groups.
  - Affected assets in incidents—table displaying the number of assets associated with unclosed incidents. The table lists assets and their importance level, which is calculated based on the categories the asset belongs to together with all parent categories up to the root of the hierarchy.
  - Affected users in incidents—users associated with unclosed incidents.
  - Affected asset categories in incidents—chart displaying the number of asset categories associated with unclosed incidents. For each asset linked to an incident, the importance level of an asset is calculated based on the categories the asset belongs to, including all parent categories up to the root of the hierarchy.
  - Active incidents by tenant—number of unclosed incidents, grouped by tenant.

  Unclosed incidents are incidents that are not in the **Done** status category.

- **Network Overview** layout (Network activity overview):
  - Netflow top internal IPs—total volume of netflow traffic received by the asset, in bytes. The data is grouped by internal IP addresses of assets.

    The widget displays up to 10 IP addresses.
  - Netflow top external IPs—total volume of netflow traffic received by the asset, in bytes. The data is grouped by external IP addresses of assets.
  - Netflow top hosts for remote control—number of events associated with access attempts to one of the following ports: 3389, 22, 135. The data is grouped by asset name.
  - Netflow total bytes by internal ports—number of bytes sent to internal ports of assets. The data is grouped by port number.
  - Top Log Sources by Events count—top 10 sources from which the greatest number of events was received.

- **NGFW Aggregated Dashboards** layout:
  - NGFW. Firewall Sessions & Actions—number of Firewall profiles actions which were triggered and for which the logging option was set.
  - NGFW. DNS Security Sessions & Actions—number of DNS Security profiles actions which were triggered and for which the logging option was set.
  - NGFW. Web Control Sessions & Actions—number of Web Control profiles actions which were triggered and for which the logging option was set.
  - NGFW. IDPS Sessions & Actions—number of IDPS profiles actions which were triggered and for which the logging option was set.
  - NGFW. Anti-Virus Sessions & Actions—number of Anti-Virus profiles actions which were triggered and for which the logging option was set.
- **NGFW IDPS & File Web Antivirus** layout:
  - NGFW. IDPS. Top-5 Threats Name—five most frequent threats detected by IDPS profiles.
  - NGFW. IDPS. Top-5 MITRE Tactics & Techniques—five most frequent MITRE ATT&CK tactics and techniques detected by IDPS profiles.
  - NGFW. IDPS. Top-5 Threats Type—five most frequent types of threat detected by IDPS profiles.
  - NGFW. IDPS. Top-5 Destination IP Addresses With the Most Blocked Web Traffic—five destination IP addresses with the largest volume of web traffic which is blocked by IDPS profiles.
  - NGFW. IDPS. Top-5 Source IP Addresses With the Most Blocked Web Traffic—five source IP addresses with the largest volume of web traffic which is blocked by IDPS profiles.
  - NGFW. File Web Antivirus. Top-5 Threats Name—five most frequent threats detected by Antivirus profiles.
  - NGFW. File Web Antivirus. Top-5 Source IP Addresses With the Most Blocked Web Traffic—five destination IP addresses with the largest volume of web traffic which is blocked by Antivirus profiles.
  - NGFW. File Web Antivirus. Top-5 Destination IP Addresses With the Most Blocked Web Traffic—five source IP addresses with the largest volume of web traffic which is blocked by Antivirus profiles.
- **NGFW DNS Security** layout:
  - NGFW. DNS Security. Top-5 Threats for Blocked Addresses—five most frequent threats from the IP addresses blocked by DNS Security profiles.
  - NGFW. DNS Security. Top-5 Threats for Redirected Addresses—five most frequent threats from the IP addresses redirected by DNS Security profiles.
  - NGFW. DNS Security. Top-5 Source IP Addresses With the Most Blocked Web Traffic—five source IP addresses with the largest volume of web traffic which is blocked by DNS Security profiles.
  - NGFW. DNS Security. Top-5 Redirected Hostnames—five domain names most frequently redirected by DNS Security profiles.

- NGFW. DNS Security. Top-5 Destination IP Addresses With the Most Blocked Web Traffic—five destination IP addresses with the largest volume of web traffic which is blocked by DNS Security profiles.
- NGFW. DNS Security. Top-5 Source IP Addresses With the Most Redirected Web Traffic—five source IP addresses most frequently redirected by DNS Security profiles.

- **NGFW Web Control & URL Web Antivirus** layout:
  - NGFW. URL Web Antivirus. Top-5 Web Categories for Blocked URLs—five most frequent categories of URLs blocked by Web Antivirus profiles.
  - NGFW. URL Web Antivirus. Top-5 Destination IP Addresses With the Most Blocked Web Traffic—five destination IP addresses with the largest volume of web traffic blocked by Web Antivirus profiles.
  - NGFW. Web Control. Top-5 Categories of Sites With the Most Warning Web Traffic—five categories of websites for which the warning from Web Control profiles is most frequently displayed.
  - NGFW. Web Control. Top-5 Categories of Sites With the Most Blocked Web Traffic—five categories of websites with the largest volume of web traffic blocked by Web Antivirus profiles.
  - NGFW. Web Control. Top-5 Destination IP Addresses With the Most Warning Web Traffic—five destination IP addresses for which the warning from Web Control profiles is most frequently displayed.
  - NGFW. Web Control. Top-5 Destination IP Addresses With the Most Blocked Web Traffic—five destination IP addresses with the largest volume of web traffic blocked by Web Control profiles.
  - NGFW. Web Control. Top-5 Web Control Category Sources for Blocked Addresses—five sources from which the categories of blocked URLs are most frequently obtained.
  - NGFW. Web Control. Top-5 Web Control Category Sources for Redirected Addresses—five sources from which the categories of redirected URLs are most frequently obtained.

- **NGFW Firewall** layout:
  - NGFW. Top-5 Application Services by Sessions—five most used application services by number of sessions.
  - NGFW. Top-5 Application Services by Traffic, MB—five most used application services by web traffic volume (MB).
  - NGFW. Top-5 Application Protocols by Sessions—five most used application protocols by number of sessions.
  - NGFW. Top-5 Application Protocols by Traffic, MB—five most used application protocols by web traffic volume (MB).
  - NGFW. Top-5 Client Applications by Sessions—five most used client applications by number of sessions.
  - NGFW. Top-5 Client Applications by Traffic, MB—five most used client applications by web traffic volume (MB).
  - NGFW. Top-5 Source IP by Traffic, MB—five source IP addresses with the largest volume of web traffic (MB).
  - NGFW. Top-5 Destination IP by Traffic, MB—five destination IP addresses with the largest volume of web traffic (MB).

- NGFW. Top-5 Web Sites by Sessions—five websites with the largest number of sessions.
- NGFW. Top-5 Web Sites by Traffic, MB—five websites with the largest volume of web traffic (MB).

The default refresh period for predefined layouts is **Never**. You can edit these layouts as needed.

About the Dashboard-only mode

You can configure the Dashboard-only mode for employees who do not manage the network but who want to view the network protection statistics in Open Single Management Platform (for example, a top manager). When a user has this mode enabled, only a dashboard with a predefined set of widgets is displayed to the user. Thus, he or she can monitor the statistics specified in the widgets, for example, the protection status of all managed devices, the number of recently detected threats, or the list of the most frequent threats in the network.

When a user works in the Dashboard-only mode, the following restrictions are applied:

- The main menu is not displayed to the user, so he or she cannot change the network protection settings.
- The user cannot perform any actions with widgets, for example, add or hide them. Therefore, you need to put all widgets required for the user on the dashboard and configure them, for instance, set the rule of counting objects or specify the time interval.

You cannot assign the Dashboard-only mode to yourself. If you want to work in this mode, contact a system administrator, Managed Service Provider (MSP), or a user with the **Modify object ACLs** right in the **General features: User permissions** functional area.

Configuring the Dashboard-only mode

Before you begin to configure the Dashboard-only mode, make sure that the following prerequisites are met:

- You have the **Modify object ACLs** right in the **General features: User permissions** functional area. If you do not have this right, the tab for configuring the mode will be missing.
- The user has the **Read** right in the **General features: Basic functionality** functional area.

> If a hierarchy of Administration Servers is arranged in your network, for configuring the Dashboard-only mode go to the Server where the user account is available on the **Users** tab of the **Users & roles** → **Users & groups** section. It can be a primary server or physical secondary server. It is not possible to adjust the mode on a virtual server.

To configure the Dashboard-only mode:

1. In the main menu, go to **Users & roles** → **Users & groups**, and then select the **Users** tab.
2. Click the user account name for which you want to adjust the dashboard with widgets.
3. In the account settings window that opens, select the **Dashboard** tab.

   On the tab that opens, the same dashboard is displayed for you as for the user.

4. If the **Display the console in Dashboard-only mode** option is enabled, switch the toggle button to disable it.

   When this option is enabled, you are also unable to change the dashboard. After you disable the option, you can manage widgets.

5. Configure the dashboard appearance. The set of widgets prepared on the **Dashboard** tab is available for the user with the customizable account. He or she cannot change any settings or size of the widgets, add, or remove any widgets from the dashboard. Therefore, adjust them for the user, so he or she can view the network protection statistics. For this purpose, on the **Dashboard** tab you can perform the same actions with widgets as in the **Monitoring** → **Dashboard** section:

   - Add new widgets to the dashboard.
   - Hide widgets that the user doesn't need.
   - Move widgets into a specific order.
   - Change the size or appearance of widgets.
   - Change the widget settings.

6. Switch the toggle button to enable the **Display the console in Dashboard-only mode** option.

   After that, only the dashboard is available for the user. He or she can monitor statistics but cannot change the network protection settings and dashboard appearance. As the same dashboard is displayed for you as for the user, you are also unable to change the dashboard.

   If you keep the option disabled, the main menu is displayed for the user, so he or she can perform various actions in Open Single Management Platform, including changing security settings and widgets.

7. Click the **Save** button when you finish configuring the Dashboard-only mode. Only after that will the prepared dashboard be displayed to the user.

8. If the user wants to view statistics of supported Kaspersky applications and needs access rights to do so, configure the rights for the user. After that, Kaspersky applications data is displayed for the user in the widgets of these applications.

Now the user can log in to Open Single Management Platform under the customized account and monitor the network protection statistics in the Dashboard-only mode.

Administration and protection reports

This section describes how to use reports, manage custom report templates, use report templates to generate new reports, and create report delivery tasks.

Using reports

The Reports feature allows you to get detailed numerical information about the security of your organization's network, save this information to a file, send it by email, and print it.

Reports are available in the OSMP Console, in the **Monitoring** section, by clicking **Reports**.

By default, reports include information for the last 30 days.

Open Single Management Platform has a default set of reports for the following categories:

- **Protection status**
- **Deployment**
- **Updating**
- **Threat statistics**
- **Other**

You can create custom report templates, edit report templates, and delete them.

You can create reports that are based on existing templates, export reports to files, and create tasks for report delivery.

Report template

Report templates are used to specify the conditions and data that a report is to be based on, as well as the reporting period. Users with the General administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, edit, or delete report templates. Reports that were generated using report templates are displayed in the **Generated reports** tab.

Report templates are available in the **Templates** tab of the **Reports** section, where the table of existing templates is displayed. The table contains the following information:

- **Name** is the name of the report template.

  You can sort report templates by clicking the sort icon ⌄ and in the column header and selecting **Ascending** or **Descending**.

  You can also search report templates by using the **Search** field above the table.

  > Regular expressions are used when searching for report templates.

- **Tenant**—the name of the tenant that owns the report template.
- **Updated** is the date and time of the last update of a resource template.

  You can sort the table by this column by clicking the sort icon ⌄ and in the column header and selecting **Ascending** or **Descending**.

- **Created by**—the name of the user who created the report template.
- **Description** is optional information about the report template.
- **Tags** to help find the report template. You can search for report templates by clicking the sort icon ⌄ in the column header, by specifying a value to search for in the **Search** field, or by selecting tags using check boxes.
- **Schedule**—the rate at which reports must be generated using the template. If the report schedule was not configured, the `disabled` value is displayed.
- **Last report**—the date and time when the last report was generated based on the report template.

- **Package name** is the name of the package of tools for processing events.
- **Send by email** indicates that email notification is configured for generated reports.

If necessary, you can customize the columns by clicking the gear icon ⚙.

After selecting a report template in the list, you can click the menu icon ⋮ to open a drop-down list of available actions:

- **Show dependencies** opens the dependency graph window for the selected report template. If necessary, in the graph window, you can download dependency data in SVG format and configure filters to be applied to dependencies displayed on the graph.
- **View versions history** displays the version history of a report template. If necessary, in the sidebar, you can compare the versions and restore a previous version.
- **Run report** generates a report. The generated reports are displayed on the **Generated reports** tab.
- **View all generated reports** displays a list of all reports generated for the selected report template. If necessary, you can save the selected report in one of the supported formats or delete the report.
- **Edit schedule** lets you configure the reporting schedule and select users that you want to receive email notifications about generated reports.

You can also view the list of report templates available to you in the **Resources** → **List** section. To do this, select **Report template** in the **Resource kind** column.

Creating a report template

To create a report template:

1. In the main menu, go to **Monitoring** → **Reports**.
2. Click **Add**.

   The New report template wizard starts. Proceed through the wizard by using the **Next** button.
3. Enter the report name and select the report type.
4. On the **Scope** step of the wizard, select the set of client devices (administration group, device selection, selected devices, or all networked devices) whose data will be displayed in reports that are based on this report template.
5. On the **Reporting period** step of the wizard, specify the report period. Available values are as follows:

   - Between the two specified dates
   - From the specified date to the report creation date
   - From the report creation date, minus the specified number of days, to the report creation date

   This page may not appear for some reports.
6. Click **OK** to close the wizard.

7. Do one of the following:

- Click the **Save and run** button to save the new report template and to run a report based on it.

  The report template is saved. The report is generated.

- Click the **Save** button to save the new report template.

  The report template is saved.

You can use the new template for generating and viewing reports.

Viewing and editing report template properties

You can view and edit basic properties of a report template, for example, the report template name or the fields displayed in the report.

To view and edit properties of a report template:

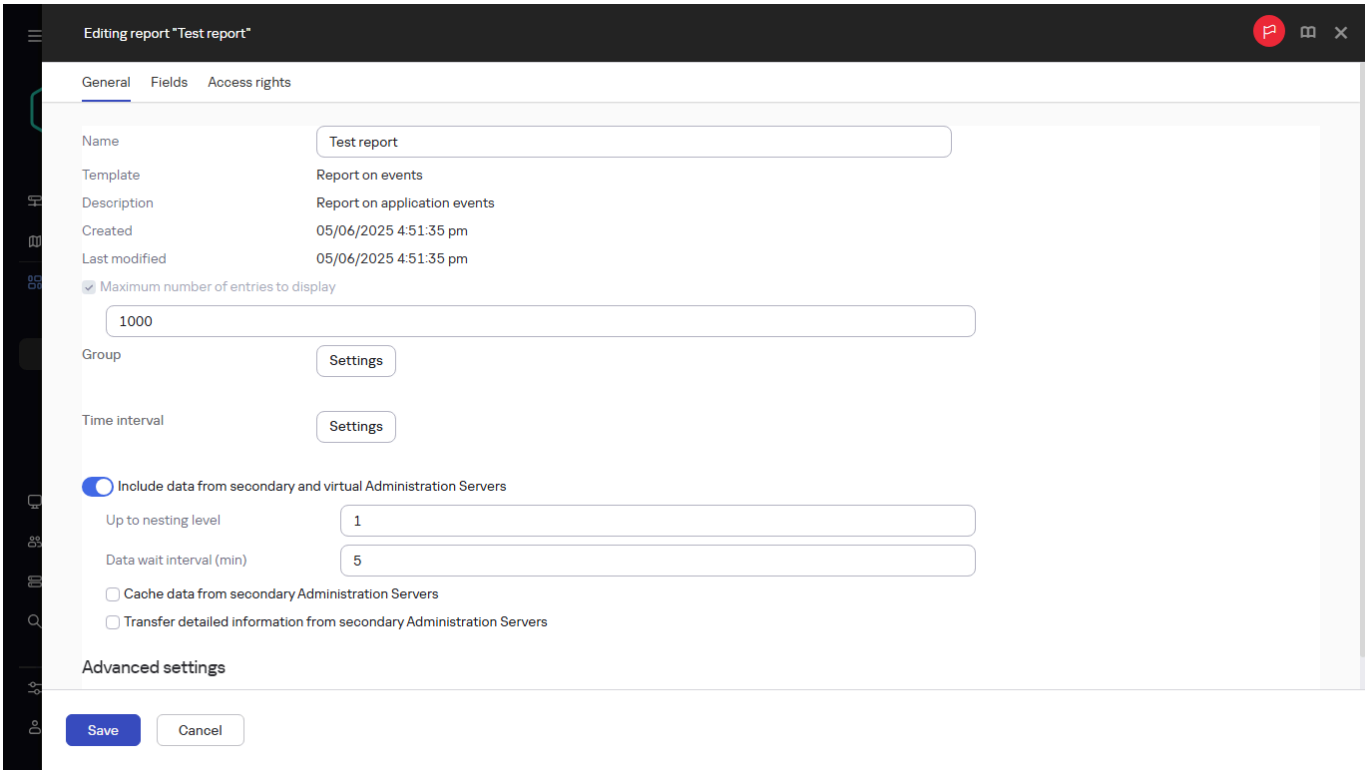
1. In the main menu, go to **Monitoring** → **Reports**.

2. Select the check box next to the report template whose properties you want to view and edit.

   As an alternative, you can first generate the report, and then click the **Edit** button.

3. Click the **Open report template properties** button.

   The **Editing report <Report name>** window opens with the **General** tab selected.



The report template properties

4. Edit the report template properties:

- **General** tab:

  - Report template name

  - **Maximum number of entries to display**

    If this option is enabled, the number of entries displayed in the table with detailed report data does not exceed the specified value. Note that this option does not affect the maximum number of events that you can include in the report when you export the report to a file.

    Report entries are first sorted according to the rules specified in the **Fields** → **Details fields** section of the report template properties, and then only the first of the resulting entries are kept. The heading of the table with detailed report data shows the displayed number of entries and the total available number of entries that match other report template settings.

    If this option is disabled, the table with detailed report data displays all available entries. We do not recommend that you disable this option. Limiting the number of displayed report entries reduces the load on the database management system (DBMS) and reduces the time required for generating and exporting the report. Some of the reports contain too many entries. If this is the case, you may find it difficult to read and analyze them all. Also, your device may run out of memory while generating such a report and, consequently, you will not be able to view the report.

    By default, this option is enabled. The default value is 1000.

  - **Group**

    Click the **Settings** button to change the set of client devices for which the report is created. For some types of the reports, the button may be unavailable. The actual settings depend on the settings specified during creation of the report template.

  - **Time interval**

    Click the **Settings** button to modify the report period. For some types of the reports, the button may be unavailable. Available values are as follows:

    - Between the two specified dates

    - From the specified date to the report creation date

    - From the report creation date, minus the specified number of days, to the report creation date

  - **Include data from secondary and virtual Administration Servers**

    If this option is enabled, the report includes the information from the secondary and virtual Administration Servers that are subordinate to the Administration Server for which the report template is created.

    Disable this option if you want to view data only from the current Administration Server.

    By default, this option is enabled.

  - **Up to nesting level**

    The report includes data from secondary and virtual Administration Servers that are located under the current Administration Server on a nesting level that is less than or equal to the specified value.

The default value is 1. You may want to change this value if you have to retrieve information from secondary Administration Servers located at lower levels in the tree.

- **Data wait interval (min)**

  Before generating the report, the Administration Server for which the report template is created waits for data from secondary Administration Servers during the specified number of minutes. If no data is received from a secondary Administration Server at the end of this period, the report runs anyway. Instead of the actual data, the report shows data taken from the cache (if the **Cache data from secondary Administration Servers** option is enabled), or **N/A** (not available) otherwise.

  The default value is 5 (minutes).

- **Cache data from secondary Administration Servers**

  Secondary Administration Servers regularly transfer data to the Administration Server for which the report template is created. There, the transferred data is stored in the cache.

  If the current Administration Server cannot receive data from a secondary Administration Server while generating the report, the report shows data taken from the cache. The date when the data was transferred to the cache is also displayed.

  Enabling this option allows you to view the information from secondary Administration Servers even if the up-to-date data cannot be retrieved. However, the displayed data can be obsolete.

  By default, this option is disabled.

- **Cache update frequency (h)**

  Secondary Administration Servers at regular intervals transfer data to the Administration Server for which the report template is created. You can specify this period in hours. If you specify 0 hours, data is transferred only when the report is generated.

  The default value is 0.

- **Transfer detailed information from secondary Administration Servers**

  In the generated report, the table with detailed report data includes data from secondary Administration Servers of the Administration Server for which the report template is created.

  Enabling this option slows the report generation and increases traffic between Administration Servers. However, you can view all data in one report.

  Instead of enabling this option, you may want to analyze detailed report data to detect a faulty secondary Administration Server, and then generate the same report only for that faulty Administration Server.

  By default, this option is disabled.

- **Fields** tab

  Select the fields that will be displayed in the report, and use the **Move up** button and **Move down** button to change the order of these fields. Use the **Add** button or **Edit** button to specify whether the information in the report must be sorted and filtered by each of the fields.

In the **Filters of Details fields** section, you can also click the **Convert filters** button to start using the extended filtering format. This format enables you to combine filtering conditions specified in various fields by using the logical OR operation. After you click the button, the **Convert filters** panel opens on the right. Click the **Convert filters** button to confirm conversion. You can now define a converted filter with conditions from the **Details fields** section that are applied by using the logical OR operation.

> Conversion of a report to the format supporting complex filtering conditions will make the report incompatible with the previous versions of Kaspersky Security Center (11 and earlier). Also, the converted report will not contain any data from secondary Administration Servers running such incompatible versions.

5. Click **Save** to save the changes.

6. Close the **Editing report <Report name>** window.

The updated report template appears in the list of report templates.

Deleting report templates

To delete one or several report templates:

1. In the main menu, go to **Monitoring** → **Reports**.

2. Select check boxes next to the report templates that you want to delete.

3. Click the **Delete** button.

4. In the window that opens, click **OK** to confirm your selection.

The selected report templates are deleted. If these report templates were included in the report delivery tasks, they are also removed from the tasks.

Universal report templates

A report template in KUMA is considered universal if it is created in the Shared tenant. Report templates created in other tenants are non-universal.

To manage a universal report template, the user needs the following roles:

- For creation, management, or deletion, the Manage shared resources role is needed. This role rights grant access to all shared resources.

  The CII attribute can be set by any user with the Manage shared resources role if the user's role involves managing CII resources.

- The Read shared resources role allows viewing the list of universal report templates. With such rights, the user gains read access to all shared resources.

Widgets in a universal report template are displayed based on the following:

- Widget data for tenants is displayed according to the user's access to tenants, the global tenant filter, and the configuration of the report template or widget.
- You can view the widget data when managing report template settings by clicking the **Update preview** button.
- Event widget data is received from the database in accordance with the space set settings of the user. If, when generating a report, it turns out that the user does not have access to the spaces of one or more tenants, KUMA generates an empty report and displays a message indicating that other results cannot be provided because of the lack of access to the relevant spaces.
- Widget data in the report can include or exclude CII. For access to reports in which data includes CII, the user must have the Interaction with CII role.
- Limitations for active list and context table widgets apply to universal reports.

Exporting a report to a file

You can save one or multiple reports as XML, HTML, or PDF. Open Single Management Platform allows you to export up to 10 reports to files of the specified format at the same time.

PDF format is available only if you are connected to the secondary Administration Server in OSMP Console.

To export a report to a file:

1. In the main menu, go to **Monitoring** → **Reports**.
2. Choose the reports that you want to export.

   If you choose more than 10 reports, the **Export report** button will be disabled.
3. Click the **Export report** button.
4. In the window that opens, specify the following export parameters:

   - **File name**.

     If you select one report to export, specify the report file name.

     If you select more than one report, the report file names will coincide with the name of the selected report templates.
   - **Maximum number of entries**.

     Specify the maximum number of entries included in the report file. The default value is 10,000.

     You can export a report with an unlimited number of entries. Note that if your report contains a large number of entries, the time required for generating and exporting the report increases.
   - **File format**.

     Select the report file format: XML, HTML, or PDF. If you export multiple reports, all selected reports are saved in the specified format as separate files.

PDF format is available only if you are connected to the secondary Administration Server in OSMP Console.

The wkhtmltopdf tool is required to convert a report to PDF. When you select the PDF option, secondary Administration Server checks whether the wkhtmltopdf tool is installed on the device. If the tool is not installed, the application displays a message about the necessity to install the tool on the Administration Server device. Install the tool manually, and then proceed to the next step.

5. Click the **Export report** button.

The report is saved to a file in the specified format.

Generating and viewing a report

To create and view a report:

1. In the main menu, go to **Monitoring** → **Reports**.

2. Click the name of the report template that you want to use to create a report.

   A report using the selected template is generated and displayed.

Report data is displayed according to the localization set for the Administration Server.

In the generated reports, some fonts may be displayed incorrectly on the diagrams. To resolve this issue, install the fontconfig library. Also, please check that the fonts corresponding to your operating system locale are installed in the operating system.

The report displays the following data:

- On the **Summary** tab:
  - The name and type of report, a brief description and the reporting period, as well as information about the group of devices for which the report is generated.
  - Graph chart showing the most representative report data.
  - Consolidated table with calculated report indicators.
- On the **Details** tab, a table with detailed report data is displayed.

Creating a report delivery task

You can create a task that will deliver selected reports.

To create a report delivery task:

1. In the main menu, go to **Monitoring** → **Reports**.

2. Select the check boxes next to the report templates for which you want to create a report delivery task.

3. Click the **Create delivery task** button.

   The New task wizard starts. Proceed through the wizard by using the Next button.

4. At the **New task settings** step of the wizard, enter the task name.

   The default name is **Deliver reports.** If a task with this name already exists, a sequence number (<N>) is added to the task name.

5. At the **Report configuration** step of the wizard, specify the following settings:

   a. Report templates to be delivered by the task.

   b. The report format: HTML, XLS, or PDF.

      The wkhtmltopdf tool is required to convert a report to PDF. When you select the PDF option, Administration Server checks whether the wkhtmltopdf tool is installed on the device. If the tool is not installed, the application displays a message about the necessity to install the tool on the Administration Server device. Install the tool manually, and then proceed to the next step.

   c. Whether the reports are to be sent by email, together with email notification settings.

      You can specify up to 20 email addresses. To separate email addresses, press **Enter**. You can also paste a comma-separated list of email addresses, and then press **Enter**.

   d. Whether the reports are to be saved to a folder, together with the corresponding settings.

      After you enable the **Save to a folder** option, you must specify a POSIX path to the folder. If you want to save the reports to a shared folder, you also have to select the **Specify account for access to shared folder** check box, and then specify the user account and password for accessing this folder.

      > If you select to save the reports to a shared folder, you have to ensure the access to this folder from the device with Administration Server installed. The ways to ensure the access and the tools used depend on your infrastructure.

      When saving the reports to a local folder, credentials are usually not needed since the account under which the Administration Server is running has the access to this folder. If necessary, you can specify the user credentials at the **Selecting an account to run the task** step of the wizard.

      Regardless of the folder choice, you can also select the **Overwrite older reports of the same type** check box if you want the new report file to overwrite the file that was saved in the reports folder at the previous task startup.

6. At the **Configure task schedule** step of the wizard, select the task start schedule.

   The following task schedule options are available:

   - **Manually**

     > The task does not run automatically. You can only start it manually.
     >
     > By default, this option is selected.

- **Every N minutes**

  The task runs regularly, with the specified interval in minutes, starting from the specified time on the day that the task is created.

  By default, the task runs every 30 minutes, starting from the current system time.

- **Every N hours**

  The task runs regularly, with the specified interval in hours, starting from the specified date and time.

  By default, the task runs every 6 hours, starting from the current system date and time.

- **Every N days**

  The task runs regularly, with the specified interval in days. Additionally, you can specify a date and time of the first task run. These additional options become available, if they are supported by the application for which you create the task.

  By default, the task runs every day, starting from the current system date and time.

- **Every N weeks**

  The task runs regularly, with the specified interval in weeks, on the specified day of week and at the specified time.

  By default, the task runs every Friday at the current system time.

- **Monthly**

  The task runs regularly, on the specified day of the month, at the specified time.

  In months that lack the specified day, the task runs on the last day.

  By default, the task runs on the first day of each month, at the current system time.

- **On specified days**

  The task runs regularly, on the specified days of each month, at the specified time.

  By default, the start time is 18:00, and no days of the month are selected.

  Note that you select neither a specific date in the month, nor the number of the week (first, second week of the month) but the sequence number of the weekday within a month. For example, if you set the cursor in the **Tu** cell of the **First** row, it means the task will run every first Tuesday of each new month.

  You can select several days of week.

- **On virus outbreak**

  The task runs after a *Virus outbreak* event occurs. Select application types that will monitor virus outbreaks. The following application types are available:

  - Anti-virus for workstations and file servers
  - Anti-virus for perimeter defense
  - Anti-virus for mail systems

  By default, all application types are selected.

  You may want to run different tasks depending on the security application type that reports a virus outbreak. In this case, remove the selection of the application types that you do not need.

- **On completing another task**

  The current task starts after another task completes. This option only works if both tasks are assigned to the same devices. For example, you may want to run the *Manage devices* task with the **Turn on the device** option and, after it completes, run the *Virus scan* task as a triggering task.

  You have to select the triggering task from the table and the status with which this task must complete (**Completed successfully** or **Failed**).

  If necessary, you can search, sort, and filter the tasks in the table as follows:

  - Enter the task name in the search field, to search the task by its name.
  - Click the sort icon to sort the tasks by name.

    By default, the tasks are sorted in alphabetical ascending order.
  - Click the filter icon, and in the window that opens, filter the tasks by group, and then click the **Apply** button.

7. At this step of the wizard, configure other task schedule settings:

- In the **Task schedule** section, check or reconfigure the previously selected schedule and set the time interval, days of the month or week, set the virus outbreak condition or completing another task as a trigger to start the task. A start time can also be specified in this section if an applicable schedule is selected.
- In the **Additional settings** section, specify the following settings:

  - **Run missed tasks**

    This option determines the behavior of a task if a client device is not visible on the network when the task is about to start.

    If this option is enabled, the system attempts to start the task the next time the Kaspersky application is run on the client device. If the task schedule is **Manually**, **Once** or **Immediately**, the task is started immediately after the device becomes visible on the network or immediately after the device is included in the task scope.

    If this option is disabled, only scheduled tasks run on client devices. For **Manually**, **Once** and **Immediately** schedule, tasks run only on those client devices that are visible on the network. For example, you may want to disable this option for a resource-consuming task that you want to run only outside of business hours.

By default, this option is disabled.

- **Use automatically randomized delay for task starts**

  If this option is enabled, the task is started on client devices randomly within a specified time interval, that is, *distributed task start*. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  The distributed start time is calculated automatically when a task is created, depending on the number of client devices to which the task is assigned. Later, the task is always started on the calculated start time. However, when task settings are edited or the task is started manually, the calculated value of the task start time changes.

  If this option is disabled, the task starts on client devices according to the schedule.

- **Use automatically randomized delay for task starts within an interval of**

  If this option is enabled, the task is started on client devices randomly within the specified time interval. A distributed task start helps to avoid a large number of simultaneous requests by client devices to the Administration Server when a scheduled task is running.

  If this option is disabled, the task starts on client devices according to the schedule.

  By default, this option is disabled. The default time interval is one minute.

- **Stop the task if it runs longer than**

  After the specified time period expires, the task is stopped automatically, whether it is completed or not.

  Enable this option if you want to interrupt (or stop) tasks that take too long to execute.

  By default, this option is disabled. The default task execution time is 120 minutes.

8. At the **Selecting an account to run the task** step of the wizard, specify the credentials of the user account that is used to run the task.

9. If you want to modify other task settings after the task is created, at the **Finish task creation** step of the wizard, enable the **Open task details when creation is complete** option (by default, this option is enabled).

10. Click the **Finish** button to create the task and close the wizard.

    The report delivery task is created. If the **Open task details when creation is complete** option is enabled, the task settings window opens.

Detection and response reports

You can schedule the generation of process and object status reports in KUMA.

Reports are generated using *report templates*. You can find report templates in the **Reports** section on the **Templates** tab.

You can find the generated reports in the **Reports** section on the **Generated reports** tab.

> To save the generated reports in HTML and PDF formats, install the required packages on the device with the KUMA Core.

When deploying KUMA in a high availability version, the time zone of the KUMA Core server and the time in the user's browser may differ. This difference is manifested by the discrepancy between the time in reports generated by schedule and the data that the user can export from widgets. To avoid this discrepancy, it is recommended to configure the report generation schedule to take into account the difference between the users' time zone and UTC.

https://www.youtube.com/watch?v=3uapwg0Pq04

Report template

Report templates are used to specify the conditions and data that a report is to be based on, as well as the reporting period. Users with the General administrator, Tenant administrator, Tier 2 analyst, and Tier 1 analyst roles can create, edit, or delete report templates. Reports that were generated using report templates are displayed in the **Generated reports** tab.

Report templates are available in the **Templates** tab of the **Reports** section, where the table of existing templates is displayed. The table contains the following information:

- **Name** is the name of the report template.

  You can sort report templates by clicking the sort icon ⌄ and in the column header and selecting **Ascending** or **Descending**.

  You can also search report templates by using the **Search** field above the table.

  > Regular expressions are used when searching for report templates.

- **Tenant**—the name of the tenant that owns the report template.

- **Updated** is the date and time of the last update of a resource template.

  You can sort the table by this column by clicking the sort icon ⌄ and in the column header and selecting **Ascending** or **Descending**.

- **Created by**—the name of the user who created the report template.
- **Description** is optional information about the report template.
- **Tags** to help find the report template. You can search for report templates by clicking the sort icon ⌄ in the column header, by specifying a value to search for in the **Search** field, or by selecting tags using check boxes.
- **Schedule**—the rate at which reports must be generated using the template. If the report schedule was not configured, the `disabled` value is displayed.
- **Last report**—the date and time when the last report was generated based on the report template.
- **Package name** is the name of the package of tools for processing events.
- **Send by email** indicates that email notification is configured for generated reports.

If necessary, you can customize the columns by clicking the gear icon ⚙.

After selecting a report template in the list, you can click the menu icon ⋮ to open a drop-down list of available actions:

- **Show dependencies** opens the dependency graph window for the selected report template. If necessary, in the graph window, you can download dependency data in SVG format and configure filters to be applied to dependencies displayed on the graph.
- **View versions history** displays the version history of a report template. If necessary, in the sidebar, you can compare the versions and restore a previous version.
- **Run report** generates a report. The generated reports are displayed on the **Generated reports** tab.
- **View all generated reports** displays a list of all reports generated for the selected report template. If necessary, you can save the selected report in one of the supported formats or delete the report.
- **Edit schedule** lets you configure the reporting schedule and select users that you want to receive email notifications about generated reports.

You can also view the list of report templates available to you in the **Resources** → **List** section. To do this, select **Report template** in the **Resource kind** column.

Creating report template

To create report template:

1. Open OSMP Console and go to one of the following sections:
   - **Reports** → **Templates**
   - **Resources** → **List**

     If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

2. Depending on the selected section, do one of the following:

   - In the **Reports** → **Templates** section, click the **Create new** button above the table of report templates.

   - In the **Resources** → **List** section, click the **Create new** button and select **Report template** in the list of resources.

3. This opens a window; in that window, enter a unique name for the report template. The name must contain 1 to 128 Unicode characters.

   Custom report templates cannot have the same names as predefined report templates in the Main tenant.

   In the Main tenant, if you try to create a report template with the same name as a predefined report template or assign the name of a predefined report template to a custom report template, an error is returned. In tenants other than Main, you can create or duplicate a custom report template with the same name as a predefined report template. We recommend using custom names for custom report templates.

4. In the **Template tenant** drop-down list, select the tenant that will own the layout being created.

5. In the **Collect data from tenants** drop-down list, select one or more tenants. Reports are generated based on data of the selected tenants.

6. If necessary, in the **Time period** drop-down list, select the time period from which you want to get analytics in one of the following ways:

   - If you want to specify an exact date, in the calendar on the left, select the start and end date of the period and click **Apply**.

     You can select a date up to and including the current date. The date and time format depends on your browser settings. If the **Range from** or **Range to** field has a value and you have not edited the time value manually, when you select a date in the calendar, the **Range from** field is automatically populated with 00:00:00.000, and the **Date to** field with 23:59:59.999. If you have manually deleted the value in the **Range from** or **Range to** field, when you select a date in the calendar, the field is automatically populated with the current time. After you select a value in one of the fields, the focus switches to the other field. If your **Range to** is earlier than your **Range from**, this earlier value is automatically inserted into the **Range from** field.

   - If you want to specify a relative period, select one of the available periods in the **Relative period** list on the right.

     The period is calculated relative to the current time.

   - If you want to specify a custom period, edit the value of the **Range from** and **Range to** fields.

     You can enter an exact date and time in the **DD.MM.YYYY HH:mm:ss.SSS** format for the Russian localization and **YYYY-MM-DD HH:mm:ss.SSS** for the English localization or a period relative to the current time as a formula. You can also combine these methods if necessary. If you do not specify milliseconds when entering the exact date, 000 is substituted automatically. If you have edited the time in the **Range from** or **Range to** fields, picking a date in the calendar does not change the time component.

     In the relative period formulas, you can use the now parameter for the current date and time and the interval parameterization language: + (only in the **Range to** field), -, / (rounding to the nearest), as well as time units: **y** (year), **M** (month), **w** (week), **d** (day), **h** (hour), **m** (minute), **s** (second). For example, you can specify the **now-5d** period to get data for the last five days, or **now/w** to get data from the beginning of the first day of the current week (00:00:00:000 UTC) to the current time (**now**).

     > The **Range from** field is required, and its value cannot exceed the value of the **Range to** field, and also cannot be earlier than 1970-01-01 (if specifying an exact date or a relative period). The **Range to** cannot be earlier than the **Range from**. If you do not specify a value in the **Range to** from field, **now** is specified automatically.

By default, the **1 day (now-1d)** relative period is selected. The bounds of the period are inclusive: for example, for the **Today** time range, events are displayed from the beginning (00:00:00:000) of the current day to the current time (**now**) inclusive, and for the **Yesterday** period, events are displayed from the beginning (00:00:00:000) of the previous day to 00:00:00:000 of the current day.

KUMA stores time values in UTC, but in the user interface time is converted to the time zone of your browser. This is relevant to the relative periods: **Today**, **Yesterday**, **This week**, and **This month**. For example, if the time zone in your browser is UTC+3, and you select **Today** as the data display period, data will be displayed for the period from 03:00:00.000 until now, not from 00:00:00.000 until now.

If you want to take your time zone into account when selecting a relative data display period, such as **Today**, **Yesterday**, **This week**, or **This month**, you need to manually add a time offset in the **Range from** and **Range to** (if a value other than now is specified) by adding or subtracting the correct number of hours. For example, if your browser's time zone is UTC+3 and you want to display data for **Yesterday**, you need to change **Range from** to **now-1d/d-3h** and **Date to** to **now/d-3h**. If you want to display data for the **Today** period, you only need to change the value in the **Range from** field to **now/d-3h**.

> If you need results up to 23:59:59:999 UTC of yesterday, you can use an SQL query with a filter by Timestamp or specify an exact date and time.

7. If necessary, in the **Retention period** field, specify how long you want to store reports that are generated according to this template.

   The default value is 3 months.

8. If necessary, select tags for the report template from the **Tags** drop-down list.

9. If necessary, add a description of the report template.

10. Use the **CII status** toggle switch to enable the display of information for assets, alerts, and incidents related to critical information infrastructure (CII). In this case, reports in which the calculated data includes CII information, can be viewed only by users in whose settings the **Access to CII facilities** check box is selected.

    If this toggle switch is disabled, widgets of the reports do not display information for assets, alerts, and incidents related to critical information infrastructure (CII) even if the user has access to CII facilities.

11. Click **Save**.

12. This opens a window; in that window, you can change the logo of the report template by clicking the **Upload a file** link. In the file selection window, select the image file for the logo. The format of the image must be JPEG, PNG, or GIF, and its size may not exceed 3 MB.

    The added logo is displayed in the report instead of KUMA logo.

13. Click the **Add widget** button and select a widget. This opens a window; in that window, configure the selected widget.

14. You can add multiple widgets. You can drag widgets around the window by the handle icon (⠿) in the upper-left corner of the widget. If you need to resize the widget, you can do so using the resize icon (⤡) that is displayed when you hover over the lower-right corner of the widget.

15. The following limitations apply to widgets with the **Pie chart**, **Bar chart**, **Line chart**, **Counter**, and **Date Histogram** chart types:

    - In `SELECT` queries, you can use extended event schema fields of `String`, `Number`, and `Float` types.

    - In `WHERE` queries, you can use all types of extended event schema fields (`String`, `Number`, `Float`, `Array of strings`, `Array of numbers`, and `Array of floats`).

For widgets with the `Table` chart type, in `SELECT` queries, you can use all types of extended event schema fields (`String`, `Number`, `Float`, `Array of strings`, `Array of numbers`, and `Array of floats`).

16. By clicking the menu ⋮ icon in the upper-right corner of the widget, you can do the following:

    - Manage the settings of widgets
    - Create copies of widgets
    - Delete widgets
    - Download widget data in CSV format

17. Click **Create**.

    The new report template is displayed on the **Reports** → **Templates** and **Resources** → **List** tabs of OSMP Console. You can run this report manually. If you want to have the reports generated automatically, you need to configure a schedule.

Configuring report schedule

To configure the report schedule:

1. Open the KUMA web interface and select **Reports** → **Templates**.
2. In the table with the list of report templates, select a report template and click the menu icon ⋮.
3. In the drop-down list, select **Edit schedule**.

   This opens the **Edit schedule** window.
4. If you want the report to be generated regularly:

   a. Enable the **Send report by schedule** toggle switch.

   b. Under **Recur every**, specify how often the reports must be generated.

      You can specify the frequency of generating reports by days, weeks, months, or years. Depending on the selected period, you should specify the time, day of the week, day of the month or the date of the report generation.

   c. In the **Time** field, enter the time when the report must be generated.

   d. In the **Timezone** drop-down list, select the time zone for the report generation time.
5. Under **Notification settings**, specify the following:

   a. In the **Notification template** drop-down list, select a notification form that you want to be sent to the user on a schedule.

   b. In the **Send to** group of settings, click **Add**.

c. This opens a window; in that window, add a group of users that you want to get the report and select a format. To do so:

- In the **Emails** drop-down list, select email addresses of report recipients.
- In the **Format** drop-down list, select the format of the report. The following formats are available: PDF, HTML, CSV, split CSV, Excel.

You can add multiple groups of report recipients by clicking the **Add group** button.

Reports are sent to the specified addresses every time you generate a report manually or KUMA generates a report automatically on schedule.

> If you want the generated reports to be sent by email, you need to configure an SMTP connection.

If recipients of an emailed report are KUMA users, they can download or view the report by clicking links in the email message. If the recipients are not KUMA users, they can follow the links but cannot log in to KUMA, so only attachments are available to them.

We recommend viewing HTML reports by clicking links in the web interface, because at some screen resolutions, the HTML report from the attachment may not be displayed correctly.

If you send an email message without attachments, the recipients can only get reports by following links, and only if they can authorize in KUMA, without restrictions on roles or tenants.

d. Click **Add**.

6. Click **Save**.

Report schedule is configured.

Editing report template

To edit report template:

1. Open the KUMA web interface and go to one of the following sections:

   - **Reports** → **Templates**
   - **Resources** → **List**

     If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

2. Depending on the selected section, do one of the following:

   - In the **Reports** → **Templates** section, click the name of the report template in the table of report templates.
   - In the **Resources** → **List** section, click the edit settings icon ( ) next to the name of the report template in the list of resources.

   This opens the **Edit template** window.

   You can also open this window on the Reports → Generated reports tab by selecting an existing report and clicking the Edit template button.

> You can also open this window in the **Reports** → **Generated reports** tab by selecting an existing report and clicking the **Edit template button**.

3. Make the necessary changes:

   - Change the name of the report template.
   - Modify the list of tenants that send data for the report.
   - Click the **Settings** button to change the time period for getting analytical data, the retention period, tags, and description of the report template.
   - Click the **Add widget** button to add a widget to the report template.
   - Click the menu icon (⋮) to view the list of reports generated by this template, or to delete the report template.
   - Change the report logo by clicking the **Upload a file** link and uploading a new logo. If the template already contains a logo, you must first delete it.
   - Change widgets positions by dragging them.
   - Resize widgets using the resize icon (⤡) that appears when you hover over a widget.
   - Click the menu icon (⋮) on the widget to manage widget settings, copy widgets, delete widgets, or download widget data in CSV format.

4. Click **Save**.

   The report template is modified and displayed on the **Reports** → **Templates** and **Resources** → **List** tabs of the KUMA web interface.

Copying report template

To create a copy of a report template:

1. Open the KUMA web interface and go to one of the following sections:

   - **Reports** → **Templates**
   - **Resources** → **List**

     If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

2. In the table, select a report template and click the **Duplicate** button above the table.

3. This opens a window; in that window, enter the name of the report template. When you copy a report template, the name of the report mentions that it is a copy.

4. In the **Template tenant** drop-down list, select the tenants that will own the report template being created.

5. In the **Collect data from tenants** drop-down list, select one or more tenants whose data you want to be represented in the report template.

6. Change the following settings as needed:

   - Time interval for generating the report.
   - Report retention period.
   - Tags of the report template.
   - Description of the report template.
   - Generation of report data with CII information.

7. Click **Save**.

   The copy of the report template is created and is displayed on the **Reports** → **Templates** and **Resources** → **List** tabs of the KUMA web interface.

Deleting report template

To delete report template:

1. Open the KUMA web interface and go to one of the following sections:

   - **Reports** → **Templates**
   - **Resources** → **List**

     If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

2. In the table of report templates, select the check box of a report template and click the **Delete** button above the table.

   A confirmation window opens.

3. Do one of the following:

   - If you want to delete only the report template, click the **Delete** button.
   - If you want to delete a report template and all the reports that were generated using that template, click the **Delete with reports** button.

The report template is deleted.

Exporting or importing a report template

To export a report template:

1. Open the KUMA web interface and go to one of the following sections:

   - **Reports** → **Templates**
   - **Resources** → **List**

     If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

   This opens a window with a list of all report templates available to you in KUMA.

2. Use the check box to select the report template to be exported.
3. Click the **Export** button above the table of report templates.
4. In the sidebar that is displayed:

   a. Enter the password for the file with exported resources. The password must satisfy the following requirements:

      - At least 16 characters.
      - At least one lowercase character.
      - At least one uppercase character.
      - At lease one numeral.
      - At least one special character.

   b. In the **Tenant** drop-down list, select the tenant from which you want to export the selected resources.

   c. Select check boxes to specify the resources that you want to export.

      If you select the **All resources** check box, all resources in folders of the selected tenant are exported.

5. Click the **Export** button.

Consider the following when exporting report templates:

- Report templates are exported without tenants, storage, and space information because these parameters are configured individually for each KUMA deployment.
- Report template widgets are also exported without tenant, storage, and space information.

To import a report template:

1. Open the KUMA web interface and go to one of the following sections:

   - **Reports** → **Templates**

   - **Resources** → **List**

     If you have selected this section, for the convenience of viewing objects, you can select **Report template** in the **Resource kind** column of the table.

   This opens a window with a list of all report templates available to you in KUMA.

2. Use the check box to select the report template to be imported.

3. Click the **Import** button above the table of resources.

4. In the sidebar, in the **Tenant** drop-down list, select the tenant to which you want to import the report template.

5. Select the import source: **File** or **Repository**.

6. Depending on the selected import source, follow these steps:

   a. If you selected **File** as the import source, follow these steps:

      - Specify the password for the imported file.

      - Click the **Select file** button to select the file that you want to import.

      - Select the check boxes of the resources that you want to import.

   b. If you selected **Repository** as the import source, select one or more repository packages in the displayed table.

7. Click the **Import** button.

8. If necessary, resolve conflicts between the resources imported from the file and the existing resources.

Consider the following when importing report templates:

- If a report template is imported into any tenant, the tenant of the imported report template is specified as the resource tenant and the data source tenant.

- After importing resources, you need to specify spaces for widgets with events and if necessary, expand the list of tenants.

Universal report templates

A report template in KUMA is considered universal if it is created in the Shared tenant. Report templates created in other tenants are non-universal.

To manage a universal report template, the user needs the following roles:

- For creation, management, or deletion, the Manage shared resources role is needed. This role rights grant access to all shared resources.

  The CII attribute can be set by any user with the Manage shared resources role if the user's role involves managing CII resources.

- The Read shared resources role allows viewing the list of universal report templates. With such rights, the user gains read access to all shared resources.

Widgets in a universal report template are displayed based on the following:

- Widget data for tenants is displayed according to the user's access to tenants, the global tenant filter, and the configuration of the report template or widget.

- You can view the widget data when managing report template settings by clicking the **Update preview** button.

- Event widget data is received from the database in accordance with the space set settings of the user. If, when generating a report, it turns out that the user does not have access to the spaces of one or more tenants, KUMA generates an empty report and displays a message indicating that other results cannot be provided because of the lack of access to the relevant spaces.

- Widget data in the report can include or exclude CII. For access to reports in which data includes CII, the user must have the Interaction with CII role.

- Limitations for active list and context table widgets apply to universal reports.

Generated reports

All reports are generated using report templates. Generated reports are available in the **Generated reports** tab of the **Reports** section and are displayed in the table with the following information:

- **Name**—the name of the report template.

  You can sort report templates by clicking the sort icon ⌄ in the column header and selecting **Ascending** or **Descending**.

  You can also search for report templates by name using the **Search** field above the table.

- **Time period**—the time period for which the report analytics were extracted.

- **Last report**—date and time when the last report was generated.

- You can sort the table by this column by clicking the sort icon ⌄ in the column header and selecting **Ascending** or **Descending**.

- **Tenant**—name of the tenant that owns the report.

- **User**—name of the user who generated the report manually. If the report was generated by schedule, the value is blank. If the report was generated in KUMA lower than 2.1, the value is blank.

If necessary, you can customize the columns by clicking the gear icon ⚙.

After selecting a report in the list of generated reports, you can:

- Create a report.
- Open and view a report.
- Save a generated report in a certain format. Available formats: HTML, PDF, CSV, split CSV, Excel. By default, 250 rows are displayed in all formats. The maximum number of table rows in PDF and HTML formats is 500. If you want to output more than 500 rows in a report, set your value for the LIMIT parameter in the SQL query and save the report in CSV format.
- Manage the settings of a report template.
- Delete a report.

Viewing reports

To open report:

1. Open the KUMA web interface and select **Reports** → **Generated reports**.
2. In the table of reports, click the name of the generated report.

   The new browser window opens with the widgets displaying report analytics. If a widget displays data on events, alerts, incidents, or active lists, you can click its header to open the corresponding section of the KUMA web interface with an active filter and/or search query that is used to display data from the widget. Default values are applied to widgets.

   To view the full data, download the report in the CSV format with the specified settings from the request.

   If you have enabled the **Show total** toggle switch in the widget settings, the widget displays the **Total** value, but when you attach or upload a report in CSV or extended CSV format, the **Total** value is not displayed. To display the **Total** value in the report, you can use Excel formulas or copy and paste the total value from the widget, or use PDF or HTML attachments.
3. If necessary, save the report in the selected format by clicking the **Save as** button.

Generating reports

You can generate report manually or configure a schedule to have it generated automatically.

To generate report manually:

1. Open the KUMA web interface and go to the **Reports** section.
2. Do one of the following:

   a. On the **Generated reports** tab:

      - Select the check box of a report in the table of reports.
      - Click the **Generate report** button above the table.

b. On the **Templates** tab:

- Select a report in the table of report templates by selecting its check box.
- Click the menu icon ⋮ and select **Generate report** from the drop-down list.

Within 15 seconds, the report is generated and displayed on the Reports → Generated reports tab.

You can also configure a reporting schedule for automatic generation of reports.

Saving reports

To save the report in the desired format:

1. Open the KUMA web interface and select **Reports** → **Generated reports**.
2. Select the check box of a generated report in the table of generated reports.
3. In the **Save as** drop-down list above the table, select a format: HTML, PDF, CSV, split CSV, Excel.
   The report is saved to the download folder configured in your browser.

You can also save the report in the desired format when you view it.

Updating reports

To update a generated report:

1. Open the KUMA web interface and go to the **Reports** → **Generated reports** section.
2. In the table of generated reports, select a report.
3. Click the **Generate report** button above the table.
   The report is updated within 15 seconds and becomes available in the **Reports** → **Generated reports** section.

Deleting reports

To delete a report:

1. Open the KUMA web interface and select **Reports** → **Generated reports**.
2. In the table of generated reports, select a report.
3. Click the **Delete** button above the table.
   A confirmation window opens.
4. Click **Delete**.
   The selected report is deleted.

Events and event selections

This section provides information about events and event selections, about the types of events that occur in Open Single Management Platform components, and about managing frequent events blocking.

About events in Open Single Management Platform

Open Single Management Platform allows you to receive information about events that occur during the operation of Administration Server and Kaspersky applications installed on managed devices. Information about events is saved in the Administration Server database.

Events by type

In Open Single Management Platform, there are the following types of events:

- General events. These events occur in all managed Kaspersky applications. An example of a general event is Virus outbreak. General events have strictly defined syntax and semantics. General events are used, for instance, in reports and dashboards.
- Managed Kaspersky applications-specific events. Each managed Kaspersky application has its own set of events.

Events by source

You can view the full list of the events that can be generated by an application on the **Event configuration** tab in the application policy. For Administration Server, you can additionally view the event list in the Administration Server properties.

Events can be generated by the following applications:

- Open Single Management Platform components:
  - Administration Server
  - Network Agent
- Managed Kaspersky applications

  For details about the events generated by Kaspersky managed applications, please refer to the documentation of the corresponding application.

Events by importance level

Each event has its own importance level. Depending on the conditions of its occurrence, an event can be assigned various importance levels. There are four importance levels of events:

- A *critical event* is an event that indicates the occurrence of a critical problem that may lead to data loss, an operational malfunction, or a critical error.
- A *functional failure* is an event that indicates the occurrence of a serious problem, error, or malfunction that occurred during operation of the application or while performing a procedure.
- A *warning* is an event that is not necessarily serious, but nevertheless indicates a potential problem in the future. Most events are designated as warnings if the application can be restored without loss of data or functional capabilities after such events occur.
- An *info* event is an event that occurs for the purpose of informing about successful completion of an operation, proper functioning of the application, or completion of a procedure.

Each event has a defined storage term, during which you can view or modify it in Open Single Management Platform. Some events are not saved in the Administration Server database by default because their defined storage term is zero. Only events that will be stored in the Administration Server database for at least one day can be exported to external systems.

Events of Open Single Management Platform components

Each Open Single Management Platform component has its own set of event types. This section lists types of events that occur in Kaspersky Security Center Administration Server and Network Agent. Types of events that occur in Kaspersky applications are not listed in this section.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. For Administration Server, you can additionally view and configure the event list in the Administration Server properties. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Data structure of event type description

For each event type, its display name, identifier (ID), alphabetic code, description, and the default storage term are provided.

- **Event type display name**. This text is displayed in Open Single Management Platform when you configure events and when they occur.
- **Event type ID**. This numerical code is used when you process events by using third-party tools for event analysis.
- **Event type** (alphabetic code). This code is used when you browse and process events by using public views that are provided in the Open Single Management Platform database and when events are exported to a SIEM system.
- **Description**. This text contains the situations when an event occurs and what you can do in such a case.

- **Default storage term**. This is the number of days during which the event is stored in the Administration Server database and is displayed in the list of events on Administration Server. After this period elapses, the event is deleted. If the event storage term value is 0, such events are detected but are not displayed in the list of events on Administration Server. If you configured to save such events to the operating system event log, you can find them there.

  You can change the storage term for events: Setting the storage term for an event

Administration Server events

This section contains information about the events related to the Administration Server.

Administration Server critical events

The table below shows the events of Kaspersky Security Center Administration Server that have the **Critical** importance level.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. For Administration Server, you can additionally view and configure the event list in the Administration Server properties. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Administration Server critical events

Event type display name	Event type ID	Event type	Description	Default storage term
Administration Server certificate for mobile devices has expired	6129	KLSRV_EV_SRV_CERT_EXPIRED_DN	Events of this type occur when the Administration Server certificates for mobile devices and UEFI devices expires. You need to update the expired certificate.	180 days
Audit: Export to SIEM failed	5130	KLAUD_EV_SIEM_EXPORT_ERROR	Events of this type occur when exporting events to the SIEM system failed due to a connection error with the SIEM system.	180 days
Device has become unmanaged	4111	KLSRV_HOST_OUT_CONTROL	Events of this type occur if a managed device is visible on the network but has not connected to Administration Server for a specific period. Find out what prevents the proper functioning of Network Agent on the device. Possible causes include network issues and removal of Network Agent from the device.	180 days
Device status is Critical	4113	KLSRV_HOST_STATUS_CRITICAL	Events of this type occur when a managed device is assigned the *Critical* status. You can configure the conditions under which the device status is changed to *Critical*.	180 days

Event type display name	Event type ID	Event type	Description	Default storage term
License expires soon	4129	KLSRV_EV_LICENSE_SRV_EXPIRE_SOON	Events of this type occur when the commercial license expiration date is approaching. Once a day Open Single Management Platform checks whether a license expiration date is approaching. Events of this type are published 30 days, 15 days, 5 days, and 1 day before the license expiration date. This number of days cannot be changed. If the Administration Server is turned off on the specified day before the license expiration date, the event will not be published until the next day. When the commercial license expires, Open Single Management Platform provides only basic functionality. You can respond to the event in the following ways: • Make sure that a reserve license key is added to Administration Server. • If you use a subscription, make sure to renew it. An unlimited subscription is renewed automatically if it has been prepaid to the service provider by the due date.	180 days
License limit has been exceeded	4099	KLSRV_EV_LICENSE_CHECK_MORE_110	Once a day Open Single Management Platform checks whether a licensing limit is exceeded. Events of this type occur when Administration Server detects that some licensing limits are exceeded by Kaspersky applications installed on client devices and if the number of currently used licensing units covered by a single license exceeds 110% of the total number of units covered by the license. Even when this event occurs, client devices are protected. You can respond to the event in the following ways: • Look through the managed devices list. Delete devices that are not in use. • Provide a license for more devices (add a valid activation code or a key file to Administration Server). Open Single Management Platform determines the rules to generate events when a licensing limit is exceeded.	180 days
Limited functionality mode	4130	KLSRV_EV_LICENSE_SRV_LIMITED_MODE	Events of this type occur when Open Single Management Platform starts to operate with basic functionality, without Vulnerability and patch management and without Mobile Device Management features. Following are causes of, and appropriate responses to, the event: • License term has expired. Provide a license to use the full functionality mode of Open Single Management Platform (add a valid activation code or a key file to Administration Server). • Administration Server manages more devices than specified by the license limit. Move devices from the administration groups of an Administration Server to those of another Administration Server (if the license limit of the other Administration Server allows).	180 days

Event type display name	Event type ID	Event type	Description	Default storage term
MDM certificate has expired	4132	KLSRV_CERTIFICATE_EXPIRED	Events of this type occur when the client certificate on mobile devices expires.	180 days
The key file has been added to the denylist	4124	KLSRV_LICENSE_BLACKLISTED	Events of this type occur when Kaspersky has added the activation code or key file that you use to the denylist. Contact Technical Support for more details.	180 days
Updates for Kaspersky application modules have been revoked	4142	KLSRV_SEAMLESS_UPDATE_REVOKED	Events of this type occur if seamless updates have been revoked (*Revoked* status is displayed for these updates) by Kaspersky technical specialists; for example, they must be updated to a newer version. The event concerns Open Single Management Platform patches and does not concern modules of managed Kaspersky applications. The event provides the reason that the seamless updates are not installed.	180 days
Virus outbreak	• 26 (for File Threat Protection) • 27 (for Mail Threat Protection) • 28 (for firewall)	GNRL_EV_VIRUS_OUTBREAK	Events of this type occur when the number of malicious objects detected on several managed devices exceeds the threshold within a short period. You can respond to the event in the following ways: • Configure the threshold in the Administration Server properties. • Create a stricter policy that will be activated, or create a task that will be run, at the occurrence of this event.	

Administration Server functional failure events

The table below shows the events of Kaspersky Security Center Administration Server that have the **Functional failure** importance level.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. For Administration Server, you can additionally view and configure the event list in the Administration Server properties. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Administration Server functional failure events

Event type display name	Event type ID	Event type	Description	Default storage term
Failed to copy the updates to the specified folder	4123	KLSRV_UPD_REPL_FAIL	Events of this type occur when software updates are copied to an additional shared folder(s). You can respond to the event in the following ways: • Check whether the user account that is employed to gain access to the folder(s) has write permission. • Check whether a user name and/or a password to the folder(s) is/are changed. • Check the internet connection, as it might be the cause of the event. Follow the instructions to update databases and software modules.	180 days

Event type display name	Event type ID	Event type	Description	Default storage term
Failed to poll the cloud segment	4143	KLSRV_KLCLOUD_SCAN_ERROR	Events of this type occur when Administration Server fails to poll a network segment in a cloud environment. Read the details in the event description and respond accordingly.	Not stored
Limit of installations has been exceeded for one of the licensed applications groups	4126	KLSRV_INVLICPROD_EXCEDED	Administration Server generates events of this type periodically (every hour). Events of this type occur if in Open Single Management Platform you manage license keys of third-party applications and if the number of installations has exceeded the limit set by the license key of the third-party application. You can respond to the event in the following ways: • Look through the managed devices list. Delete the third-party application from devices on which the application is not in use. • Use a third-party license for more devices. You can manage license keys of third-party applications using the functionality of licensed applications groups. A licensed applications group includes third-party applications that meet criteria set by you.	180 days
No free disk space	4107	KLSRV_DISK_FULL	Events of this type occur when the hard drive of the device on which Administration Server is installed runs out of free space. Free up disk space on the device.	180 days
No free space in the Administration Server database	4110	KLSRV_DATABASE_FULL	Events of this type occur when there is no free space in the Administration Server database. Administration Server does not function when its database has reached its capacity and when further recording to the database is not possible. Following are the causes of this event, depending on the DBMS that you use, and appropriate responses to the event: • Limit the number of events to store in the Administration Server database. • In the Administration Server database, there are too many events sent by the Application Control component. You can change the settings of the Kaspersky Endpoint Security policy relating to Application Control event storage in the Administration Server database. Review the information on DBMS selection.	180 days
Runtime error	4125	KLSRV_RUNTIME_ERROR	Events of this type occur because of unknown issues. Most often these are DBMS issues, network issues, and other software and hardware issues. Details of the event can be found in the event description.	180 days
Shared folder is not available	4108	KLSRV_SHARED_FOLDER_UNAVAILABLE	Events of this type occur if the shared folder of Administration Server is not available. You can respond to the event in the following ways: • Check whether the Administration Server (where the shared folder is located) is turned on and available. • Check whether a user name and/or a password to the folder is/are changed. • Check the network connection.	180 days

Event type display name	Event type ID	Event type	Description	Default storage term
The Administration Server database is unavailable	4109	KLSRV_DATABASE_UNAVAILABLE	Events of this type occur if the Administration Server database becomes unavailable. You can respond to the event in the following ways: • Check whether the remote server that has the DBMS installed is available. • View the DBMS logs to discover the reason for Administration Server database unavailability.	180 days

Administration Server warning events

The table below shows the events of Kaspersky Security Center Administration Server that have the **Warning** importance level.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. For Administration Server, you can additionally view and configure the event list in the Administration Server properties. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Administration Server warning events

Event type display name	Event type ID	Event type	Description	Default storage term
Administration Server certificate for mobile devices is expiring	6128	KLSRV_EV_SRV_CERT_EXPIRES_SOON	Events of this type occur when the Administration Server certificates for mobile devices and UEFI devices expires. You need to update the expired certificate.	90 days
APNs certificate expires soon	4136	KLSRV_APN_CERTIFICATE_EXPIRES_SOON	Events of this type occur when there are fewer than 14 days left before the APNs certificate expires. When the APNs certificate expires, you need to manually renew the APNs certificate and install it on an iOS MDM Server. We recommend that you schedule the APNs certificate renewal in advance of the expiration date.	Not stored
APNs certificate has expired	4135	KLSRV_APN_CERTIFICATE_EXPIRED	Events of this type occur when an APNs certificate expires. You need to manually renew the APNs certificate and install it on an iOS MDM Server.	Not stored
Audit: Test connection to SIEM server failed	5120	KLAUD_EV_SIEM_TEST_FAILED	Events of this type occur when an automatic connection test to the SIEM server failed.	90 days
Certificate has been removed	4134	KLSRV_CERTIFICATE_REMOVED	Events of this type occur when an administrator removes any type of certificate (General, Mail, VPN) for Mobile Device Management. After removing a certificate, mobile devices connected via this certificate will fail to connect to Administration Server. This event might be helpful when investigating malfunctions associated with the management of mobile devices.	90 days

Event type display name	Event type ID	Event type	Description	Default storage term
Certificate has been requested	4133	KLSRV_CERTIFICATE_REQUESTED	Events of this type occur when a certificate for Mobile Device Management fails to be automatically reissued. Following might be the causes and appropriate responses to the event: • Automatic reissue was initiated for a certificate for which the **Reissue certificate automatically if possible** option is disabled. This might be due to an error that occurred during creation of the certificate. Manual reissue of the certificate might be required. • If you use an integration with a public key infrastructure, the cause might be a missing SAM-Account-Name attribute of the account used for integration with PKI and for issuance of the certificate. Review the account properties.	90 days
Conflict of device names	4102	KLSRV_EVENT_HOSTS_CONFLICT	Events of this type occur when Administration Server considers two or more managed devices as a single device. Most often this happens when a cloned hard drive was used for software deployment on managed devices and without switching the Network Agent to the dedicated disk cloning mode on a reference device. To avoid this issue, switch Network Agent to the disk cloning mode on a reference device before cloning the hard drive of this device.	90 days
Connection to the primary Administration Server has been interrupted	4118	KLSRV_EV_MASTER_SRV_DISCONNECTED	Events of this type occur when a connection to the primary Administration Server is interrupted. Read the operating system log on the device where the primary Administration Server is installed and respond accordingly.	90 days
Connection to the secondary Administration Server has been interrupted	4116	KLSRV_EV_SLAVE_SRV_DISCONNECTED	Events of this type occur when a connection to the secondary Administration Server is interrupted. Read the operating system log on the device where the secondary Administration Server is installed and respond accordingly.	90 days
Device has remained inactive on the network for a long time	4103	KLSRV_EVENT_HOSTS_NOT_VISIBLE	Events of this type occur when a managed device shows inactivity for some time. Most often, this happens when a managed device is decommissioned. You can respond to the event in the following ways: • Manually remove the device from the list of managed devices. Specify the time interval after which the **Device has remained inactive on the network for a long time** event is created by using OSMP Console. • Specify the time interval after which the device is automatically removed from the group by using OSMP Console.	90 days

Event type display name	Event type ID	Event type	Description	Default storage term
Device status is Warning	4114	KLSRV_HOST_STATUS_WARNING	Events of this type occur when a managed device is assigned the *Warning* status. You can configure the conditions under which the device status is changed to *Warning*.	90 days
Failed to download file to device	4165	KLSRV_FILE_DOWNLOAD_FAILED	Events of this type occur in the following cases: • You try to download a file to the corporate container, but a corporate container has not been created. • There is not enough space on the device. • Information about the file failed to download from the server. • The file contents failed to download from the server. • The folder could not be created on the device. • The file access permission is missing on the personal device. • The file access permission is missing in the corporate container.	90 days
Failed to send the FCM message to the FCM server	4140	KLSRV_GCM_GENERAL_ERROR	Events of this type occur due to unexpected errors on the Administration Server side when working with the Google Firebase Cloud Messaging HTTP protocol. Read the details in the event description and respond accordingly. If you cannot find the solution to an issue on your own, we recommend that you contact Kaspersky Technical Support.	90 days
Failed to send the FCM message to the mobile device	4138	KLSRV_GCM_DEVICE_ERROR	Events of this type occur when Mobile Device Management is configured to use Google Firebase Cloud Messaging (FCM) for connecting to managed mobile devices with an Android operating system and FCM Server fails to handle some of the requests received from Administration Server. It means that some of the managed mobile devices will not receive a push notification. Read the HTTP code in the details of the event description and respond accordingly. For more information on the HTTP codes received from FCM Server and related errors, please refer to the Google Firebase service documentation (see chapter "Downstream message error response codes").	90 days
Frequent events have been detected		KLSRV_EVENT_SPAM_EVENTS_DETECTED	Events of this type occur when Administration Server detects a frequent event on a managed device. Refer to the following section for details: Blocking frequent events.	90 days
Hard drive almost full	4105	KLSRV_NO_SPACE_ON_VOLUMES	Events of this type occur when the hard drive of the device on which Administration Server is installed almost runs out of free space. Free up disk space on the device.	90 days

Event type display name	Event type ID	Event type	Description	Default storage term
HTTP error sending the FCM message to the FCM server	4139	KLSRV_GCM_HTTP_ERROR	Events of this type occur when Mobile Device Management is configured to use Google Firebase Cloud Messaging (FCM) for connecting managed mobile devices with the Android operating system and FCM Server reverts to the Administration Server a request with a HTTP code other than 200 (OK). Following might be the causes and appropriate responses to the event: • Problems on the FCM server side. Read the HTTP code in the details of the event description and respond accordingly. For more information on the HTTP codes received from FCM Server and related errors, please refer to the Google Firebase service documentation (see chapter "Downstream message error response codes"). • Problems on the proxy server side (if you use proxy server). Read the HTTP code in the details of the event and respond accordingly.	90 days
License limit has been exceeded	4098	KLSRV_EV_LICENSE_CHECK_100_110	Once a day Open Single Management Platform checks whether a licensing limit is exceeded. Events of this type occur when Administration Server detects that some licensing limits are exceeded by Kaspersky applications installed on client devices and if the number of currently used licensing units covered by a single license constitute 100% to 110% of the total number of units covered by the license. Even when this event occurs, client devices are protected. You can respond to the event in the following ways: • Look through the managed devices list. Delete devices that are not in use. • Provide a license for more devices (add a valid activation code or a key file to Administration Server). Open Single Management Platform determines the rules to generate events when a licensing limit is exceeded.	90 days
Limit of installations will soon be reached for one of the licensed applications groups	4128	KLSRV_INVLICPROD_EXPIRED_SOON	Events of this type occur when one of the license keys for third-party applications included in a licensed applications group is about to expire. As a result, the installation limit for one of the groups will be reduced. You can respond to the event in the following ways: • If the third-party application is not in use on some of the managed devices, delete the application from these devices. • If you expect that the number of installations for the third-party application will exceed the allowed maximum in the near future, consider obtaining a third-party license for a greater number of devices in advance. You can manage license keys of third-party applications using the functionality of licensed applications groups.	90 days
New updates for Kaspersky application modules have been registered	4141	KLSRV_SEAMLESS_UPDATE_REGISTERED	Events of this type occur when Administration Server registers new updates for the Kaspersky software installed on managed devices that require approval to be installed. Approve or decline the updates by using Kaspersky Security Center Web Console.	90 days

Event type display name	Event type ID	Event type	Description	Default storage term
No free space in the Administration Server database	4106	KLSRV_NO_SPACE_IN_DATABASE	Events of this type occur if space in the Administration Server database is too limited. If you do not remedy the situation, soon the Administration Server database will reach its capacity and Administration Server will not function. Following are the causes of this event, depending on the DBMS that you use, and the appropriate responses to the event. • Do not limit the number of events to store in the Administration Server database • Reduce the list of events to store in the Administration Server database Review the information on DBMS selection.	90 days
The limit on the number of events in the database is exceeded, deletion of events has started	4145	KLSRV_EVP_DB_TRUNCATING	Events of this type occur when deletion of old events from the Administration Server database has started after the Administration Server database capacity is reached. You can respond to the event in the following ways: • Change the maximum number of events stored in the Administration Server database • Reduce the list of events to store in the Administration Server database	Not stored
The limit on the number of events in the database is exceeded, the events have been deleted	4146	KLSRV_EVP_DB_TRUNCATED	Events of this type occur when old events have been deleted from the Administration Server database after the Administration Server database capacity is reached. You can respond to the event in the following ways: • Change the allowed maximum number of events to be stored in the Administration Server database • Reduce the list of events to store in the Administration Server database	Not stored

Administration Server informational events

The table below shows the events of Kaspersky Security Center Administration Server that have the **Info** importance level.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. For Administration Server, you can additionally view and configure the event list in the Administration Server properties. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Administration Server informational events

Event type display name	Event type ID	Event type	Description	Default storage term
Administration Server certificate for mobile devices is being renewed	6127	KLSRV_EV_SRV_CERT_RENEWED	This event occurs when the Administration Server certificate has been renewed.	30 days
Audit: Connection to Administration Server has been terminated	4151	KLAUD_EV_SERVERDISCONNECT		30 days

Event type display name	Event type ID	Event type	Description	Default storage term
Audit: Connection to the Administration Server has been established	4147	KLAUD_EV_SERVERCONNECT	Events of this type occur when a user connects to Administration Server by using OSMP Console. These events include the IP address of the device where the OSMP Console Server is installed.	30 days
Audit: Encryption keys imported/exported	5100	KLAUD_EV_DPEKEYSEXPORT	For example, this event occurs during migration.	30 days
Audit: Group settings have been modified	4149	KLAUD_EV_ADMGROUP_CHANGED	Events of this type occur when a security group has been edited.	30 days
Audit: Object has been modified	4148	KLAUD_EV_OBJECTMODIFY	This event tracks changes in the following objects: • Administration group • Security group • User • Package • Task • Policy • Server • Virtual Server	30 days
Audit: Object properties have been modified	4152	KLAUD_EV_OBJECTPROPMODIFIED	This event tracks changes in the following properties: • User • License • Server • Virtual server	30 days
Audit: Object status has changed	4150	KLAUD_EV_TASK_STATE_CHANGED	For example, this event occurs when a task has failed with an error.	30 days
Audit: Test connection to SIEM server succeeded	5110	KLAUD_EV_SIEM_TEST_SUCCESS	This event occurs when a test connection to the SIEM server succeeded.	30 days
Audit: User permissions have been modified	4153	KLAUD_EV_OBJECTACLMODIFIED	This event occurs when user permissions have been modified	30 days
Connection to the primary Administration Server has been established	4117	KLSRV_EV_MASTER_SRV_CONNECTED		30 days
Connection to the secondary Administration Server has been established	4115	KLSRV_EV_SLAVE_SRV_CONNECTED	Refer to the following article for details: Creating a hierarchy of Administration Servers: adding a secondary Administration Server.	30 days
Databases have been updated	4144	KLSRV_UPD_BASES_UPDATED	Events of this type occur when the **Download updates to the Administration Server repository** task finishes updating databases.	30 days
Device has been automatically added to the group	4101	KLSRV_EVENT_HOSTS_NEW_REDIRECTED	Events of this type occur when devices have been assigned to a group according to device moving rules.	30 days
Device has been automatically moved according to a rule	1074	KLSRV_HOST_MOVED_WITH_RULE_EX	Events of this type occur when devices have been moved to administration groups by using device moving rules.	30 days

Event type display name	Event type ID	Event type	Description	Default storage term
Device has been removed from the group: inactive on the network for a long time	4104	KLSRV_INVISIBLE_HOSTS_REMOVED	Events of this type occur when devices have been automatically removed from a group for inactivity.	30 days
FCM Instance ID has changed on this mobile device	4137	KLSRV_GCM_DEVICE_REGID_CHANGED	Events of this type occur when the Firebase Cloud Messaging token has changed on the device. For information on the FCM token rotation, please refer to the Firebase service documentation.	30 days
File deleted from Administration Server	4163	KLSRV_FILE_REMOVED	This event occurs when a file has been deleted from Administration Server.	30 days
File downloaded to device	4164	KLSRV_FILE_DOWNLOADED	This event occurs in the following cases: • A file has been downloaded to a device in a session. • A file has been downloaded from Administration Server.	30 days
Files have been found to send to Kaspersky for analysis	4131	KLSRV_APS_FILE_APPEARED		30 days
File uploaded to Administration Server	4162	KLSRV_FILE_UPLOADED	This event occurs when a file has been uploaded to Administration Server.	30 days
Limit of installations will soon be exceeded (more than 90% is used up) for one of the licensed applications groups	4127	KLSRV_INVLICPROD_FILLED	Events of this type occur when the number of installations for third-party applications included in a licensed applications group reaches 90% of the maximum allowed value specified in the license key properties. You can respond to the event in the following ways: • If the third-party application is not in use on some of the managed devices, delete the application from these devices. • If you expect that the number of installations for the third-party application will exceed the allowed maximum in the near future, consider obtaining a third-party license for a greater number of devices in advance. You can manage license keys of third-party applications using the functionality of licensed applications groups.	30 days
New device has been detected	4100	KLSRV_EVENT_HOSTS_NEW_DETECTED	Events of this type occur when new networked devices have been discovered.	30 days

Event type display name	Event type ID	Event type	Description	Default storage term
Over 90% of the license key is used up	4097	KLSRV_EV_LICENSE_CHECK_90	Events of this type occur when Administration Server detects that some licensing limits are close to being exceeded by Kaspersky applications installed on client devices and if the number of currently used licensing units covered by a single license constitute over 90% of the total number of units covered by the license. Even when a licensing limit is exceeded, client devices are protected. You can respond to the event in the following ways: • Look through the managed devices list. Delete devices that are not in use. • Provide a license for more devices (add a valid activation code or a key file to Administration Server). Open Single Management Platform determines the rules to generate events when a licensing limit is exceeded.	30 days
Reserve Administration Server certificate for mobile devices has been created	6126	KLSRV_EV_SRV_CERT_RESERVE_CREATED	This event occurs when an Administration Server certificate has been created.	30 days
Updates have been successfully copied to the specified folder	4122	KLSRV_UPD_REPL_OK	Events of this type occur when the **Download updates to the Administration Server repository** task finishes copying files to a specified folder.	30 days

The events **Reserve certificate created** and **Certificate renewing** are automatically published by Administration Server and displayed in Administration Console only when the operations corresponding to these events are performed for Administration Server certificates for mobile devices and UEFI devices.

Network Agent events

This section contains information about the events related to Network Agent.

Network Agent warning events

The table below shows the events of Network Agent that have the **Warning** severity level.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Network Agent warning events

Event type display name	Event type ID	Event type	Description	Default storage term
KSN Proxy has started. Failed to check KSN for availability	7718	KSNPROXY_STARTED_CON_CHK_FAILED	Events of this type occur when test connection fails for the configured KSN proxy connection.	30 days
Security issue has occurred	549	GNRL_EV_APP_INCIDENT_OCCURED	Events of this type occur when an incident has been found on a device. For example, this event occurs when the device has low disk space.	30 days
Sudoers file found doesn't match reference value	7724	KLNAG_EV_SUDOER_DIFFERENT	Events of this type occur when there is a mismatch between the sudoers file and the reference file.	30 days
Third-party software update installation has been postponed	7698	KLNAG_EV_3P_PATCH_INSTALL_SLIPPED	For example, events of this type occur when EULA for a third-party update installation is declined.	30 days
Third-party software update installation has completed with a warning	7696	KLNAG_EV_3P_PATCH_INSTALL_WARNING	Download the trace files and check the KLRI_PATCH_RES_DESC field value for details.	30 days
User management: warnings	7722	KLNAG_EV_USR_MNG_WRN	General warning event.	30 days
Warning has been returned during installation of the software module update	7701	KLNAG_EV_PATCH_INSTALL_WARNING	Download the trace files and check the KLRI_PATCH_RES_DESC field value for details.	30 days

Network Agent informational events

The table below shows the events of Network Agent that have the **Info** severity level.

For each event that can be generated by an application, you can specify notification settings and storage settings on the **Event configuration** tab in the application policy. If you want to configure notification settings for all the events at once, configure general notification settings in the Administration Server properties.

Network Agent informational events

Event type display name	Event type ID	Event type	Default storage term
Application has been installed	7703	KLNAG_EV_INV_APP_INSTALLED	30 days
Application has been uninstalled	7704	KLNAG_EV_INV_APP_UNINSTALLED	30 days
Certificate received for Web Server HTTPS port	—	WEB_SERVER_CERTIFICATE_RENEWED	30 days
Device has been authorized	7711	KLNAG_EV_NAC_HOST_AUTHORIZED	30 days
New device has been detected	7710	KLNAG_EV_NAC_DEVICE_DISCOVERED	30 days
Device has been removed	7709	KLNAG_EV_DEVICE_REMOVE	30 days
Installation of the software module update has started	7700	KLNAG_EV_PATCH_INSTALL_STARTING	30 days
KSN Proxy has started. KSN availability check has completed successfully	7719	KSNPROXY_STARTED_CON_CHK_OK	30 days
KSN Proxy has stopped	7720	KSNPROXY_STOPPED	30 days
Monitored application has been installed	7705	KLNAG_EV_INV_OBS_APP_INSTALLED	30 days
Monitored application has been uninstalled	7706	KLNAG_EV_INV_OBS_APP_UNINSTALLED	30 days
New device has been added	7708	KLNAG_EV_DEVICE_ARRIVAL	30 days

Event type display name	Event type ID	Event type	Default storage term
Root certificates installed	7727	KLNAG_EV_ROOT_CERT_INSTALLED	30 days
Root certificates removed	7729	KLNAG_EV_ROOT_CERT_REMOVED	30 days
Sudoers file successfully restored to reference value	7725	KLNAG_EV_SUDOER_RESTORED	30 days
Third-party application has been installed	7707	KLNAG_EV_INV_CMPTR_APP_INSTALLED	30 days
Third-party software update has been installed successfully	7694	KLNAG_EV_3P_PATCH_INSTALLED_SUCCESSFULLY	30 days
Third-party software update installation has started	7695	KLNAG_EV_3P_PATCH_INSTALL_STARTING	30 days
Update for software modules has been installed successfully	7699	KLNAG_EV_PATCH_INSTALLED_SUCCESSFULLY	30 days
User management: info	7721	KLNAG_EV_USR_MNG_INF	30 days
Web Server started on host	—	WEB_SERVER_STARTED	30 days
Web Server stopped on host	—	WEB_SERVER_STOPPED	30 days
Windows Desktop Sharing: Application has been started	7714	KLUSRLOG_EV_PROCESS_LAUNCHED	30 days
Windows Desktop Sharing: File has been modified	7713	KLUSRLOG_EV_FILE_MODIFIED	30 days
Windows Desktop Sharing: File has been read	7712	KLUSRLOG_EV_FILE_READ	30 days
Windows Desktop Sharing: Started	7715	KLUSRLOG_EV_WDS_BEGIN	30 days
Windows Desktop Sharing: Stopped	7716	KLUSRLOG_EV_WDS_END	30 days

Using event selections

Event selections provide an onscreen view of named sets of events that are selected from the Administration Server database. These sets of events are grouped according to the following categories:

- By importance level—**Critical events**, **Functional failures**, **Warnings**, and **Info events**
- By time—**Recent events**
- By type—**User requests** and **Audit events**

You can create and view user-defined event selections based on the settings available, in the OSMP Console interface, for configuration.

Event selections are available in the OSMP Console, in the **Monitoring** section, by clicking **Event selections**.

By default, event selections include information for the last seven days.

Open Single Management Platform has a default set of event (predefined) selections:

- Events with different importance levels:
  - **Critical events**
  - **Functional failures**

- **Warnings**
- **Informational messages**

- **User requests** (events of managed applications)
- **Recent events** (over the last week)
- Audit events.

In Kaspersky Next XDR Expert, audit events related to service operations in your OSMP Console are displayed. These events are conditioned by actions of Kaspersky specialists. These events, for example include the following: logging in to Administration Server; Administration Server ports changing; Administration Server database backup; creation, modification, and deletion of user accounts.

You can also create and configure additional user-defined selections. In user-defined selections, you can filter events by the properties of the devices they originated from (device names, IP ranges, and administration groups), by event types and severity levels, by application and component name, and by time interval. It is also possible to include task results in the search scope. You can also use a simple search field where a word or several words can be typed. All events that contain any of the typed words anywhere in their attributes (such as event name, description, component name) are displayed.

Both for predefined and user-defined selections, you can limit the number of displayed events or the number of records to search. Both options affect the time it takes Open Single Management Platform to display the events. The larger the database is, the more time-consuming the process can be.

You can do the following:

- Edit properties of event selections
- Generate event selections
- View details of event selections
- Delete event selections
- Delete events from the Administration Server database

Creating an event selection

To create an event selection:

1. In the main menu, go to **Monitoring** → **Event selections**.
2. Click **Add**.
3. In the **New event selection** window that opens, specify the settings of the new event selection. Do this in one or more of the sections in the window.
4. Click **Save** to save the changes.

   The confirmation window opens.

5. To view the event selection result, keep the **Go to selection result** check box selected.

6. Click **Save** to confirm the event selection creation.

If you kept the **Go to selection result** check box selected, the event selection result is displayed. Otherwise, the new event selection appears in the list of event selections.

Editing an event selection

To edit an event selection:

1. In the main menu, go to **Monitoring** → **Event selections**.

2. Select the check box next to the event selection that you want to edit.

3. Click the **Properties** button.

   An event selection settings window opens.

4. Edit the properties of the event selection.

   > For predefined event selections, you can edit only the properties on the following tabs: **General** (except for the selection name), **Time**, and **Access rights**.

   For user-defined selections, you can edit all properties.

5. Click **Save** to save the changes.

The edited event selection is shown in the list.

Viewing a list of an event selection

To view an event selection:

1. In the main menu, go to **Monitoring** → **Event selections**.

2. Select the check box next to the event selection that you want to start.

3. Do one of the following:

   - If you want to configure sorting in the event selection result, do the following:

     a. Click the **Reconfigure sorting and start** button.

     b. In the displayed **Reconfigure sorting for event selection** window, specify the sorting settings.

     c. Click the name of the selection.

   - Otherwise, if you want to view the list of events as they are sorted on the Administration Server, click the name of the selection.

   The event selection result is displayed.

Exporting an event selection

Open Single Management Platform allows you to save an event selection and its settings to a KLO file. You can use this KLO file to import the saved event selection both to Kaspersky Security Center Windows and Kaspersky Security Center Linux.

> Note that you can export only user-defined event selections. Event selections from the default set of Open Single Management Platform (predefined selections) cannot be saved to a file.

To export an event selection:

1. In the main menu, go to **Monitoring** → **Event selections**.

2. Select the check box next to the event selection that you want to export.

   You cannot export multiple event selections at the same time. If you select more than one selection, the **Export** button will be disabled.

3. Click the **Export** button.

4. In the opened **Save as** window, specify the event selection file name and path, and then click the **Save** button.

   The **Save as** window is displayed only if you use Google Chrome, Microsoft Edge, or Opera. If you use another browser, the event selection file is automatically saved in the **Downloads** folder.

Importing an event selection

Open Single Management Platform allows you to import an event selection from a KLO file. The KLO file contains the exported event selection and its settings.

To import an event selection:

1. In the main menu, go to **Monitoring** → **Event selections**.

2. Click the **Import** button, and then choose an event selection file that you want to import.

3. In the opened window, specify the path to the KLO file, and then click the **Open** button. Note that you can select only one event selection file.

   The event selection processing starts.

The notification with the import results appears. If the event selection is imported successfully, you can click the **View import details** link to view the event selection properties.

After a successful import, the event selection is displayed in the selection list. The settings of the event selection are also imported.

> If the newly imported event selection has a name identical to that of an existing event selection, the name of the imported selection is expanded with the **(<next sequence number>)** index, for example: **(1)**, **(2)**.

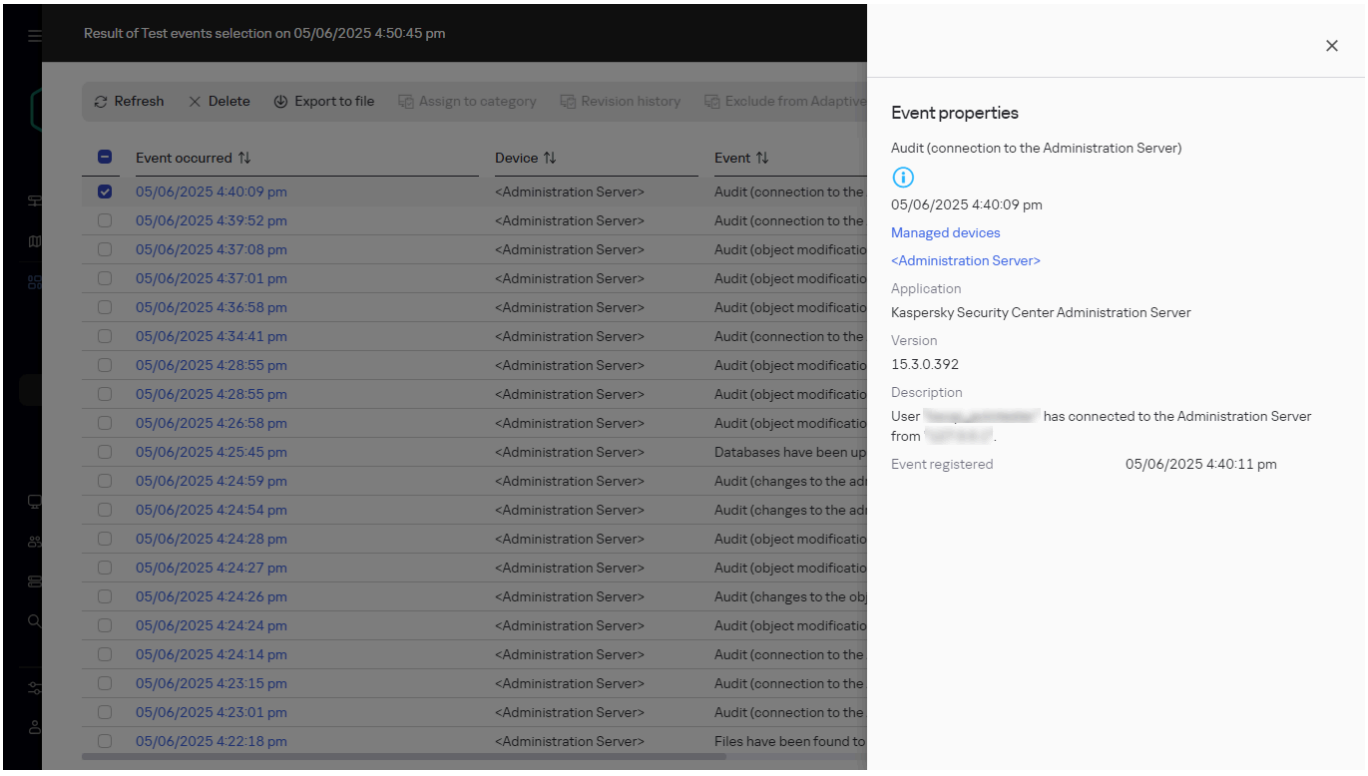
Viewing details of an event

To view details of an event:

1. Start an event selection.

2. Click the time of the required event.

   The **Event properties** window opens.



The event properties window

3. In the displayed window, you can do the following:

   - View the information about the selected event

   - Go to the device on which the event occurred

- Go to the administration group that includes the device on which the event occurred
- For an event related to a task, go to the task properties

Exporting events to a file

Open Single Management Platform allows you to save events from an event selection to a TXT file.

To export events to a file:

1. Start an event selection.
2. Select the check box next to the required event.
   You can also select several events or the entire event selection.
3. Click the **Export to file** button.

The selected event is exported to a TXT file.

Viewing an object history from an event

From an event of creation or modification of an object that supports revision management, you can switch to the revision history of the object.

To view an object history from an event:

1. Start an event selection.
2. Select the check box next to the required event.
3. Click the **Revision history** button.

The revision history of the object is opened.

Deleting events

To delete one or several events:

1. Start an event selection.
2. Select the check boxes next to the required events.
3. Click the **Delete** button.

The selected events are deleted and cannot be restored.

Deleting event selections

You can delete only user-defined event selections. Predefined event selections cannot be deleted.

To delete one or several event selections:

1. In the main menu, go to **Monitoring** → **Event selections**.
2. Select the check boxes next to the event selections that you want to delete.
3. Click **Delete**.
4. In the window that opens, click **OK**.

   The event selection is deleted.

Setting the storage term for an event

Open Single Management Platform allows you to receive information about events that occur during the operation of Administration Server and Kaspersky applications installed on managed devices. Information about events is saved in the Administration Server database. You might need to store some events for a longer or shorter period than specified by default values. You can change the default settings of the storage term for an event.

If you are not interested in storing some events in the database of Administration Server, you can disable the appropriate setting in the Administration Server policy and Kaspersky application policy, or in the Administration Server properties (only for Administration Server events). This will reduce the number of event types in the database.

The longer the storage term for an event, the faster the database reaches its maximum capacity. However, a longer storage term for an event lets you perform monitoring and reporting tasks for a longer period.

To set the storage term for an event in the database of Administration Server:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the name of the required policy.

   You can select a policy of a managed Kaspersky application, Network Agent, or Administration Server. For Administration Server, you can also configure the storage term of the events by clicking the Administration Server name.
3. Select the **Event configuration** tab.

   A list of event types related to the **Critical** section is displayed. If necessary, you can move to the **Functional failure**, **Warning**, or **Info** section.
4. In the list of event types in the right pane, click the link for the event whose storage term you want to change.

   In the **Event registration** section of the window that opens, the **Store in the Administration Server database for (days)** toggle button is enabled.

5. In the edit box below this toggle button, enter the number of days to store the event.

6. If you do not want to store an event in the Administration Server database, disable the **Store in the Administration Server database for (days)** option.

   If you configure Administration Server events in Administration Server properties window and if event settings are locked in the Kaspersky Security Center Administration Server policy, you cannot redefine the storage term value for an event.

7. Click **OK**, and then after the right pane is closed, click the **Save** button.

   The properties window of the policy is closed.

From now on, when Administration Server receives and stores the events of the selected type, they will have the changed storage term. Administration Server does not change the storage term of previously received events.

Blocking frequent events

This section provides information about managing frequent events blocking and about removing blocking of frequent events.

About blocking frequent events

A managed application, for example, Kaspersky Endpoint Security for Linux, installed on a single or several managed devices can send a lot of events of the same type to the Administration Server. Receiving frequent events may overload the Administration Server database and overwrite other events. Administration Server starts blocking the most frequent events when the number of all the received events exceeds the specified limit for the database.

Administration Server blocks the frequent events from receiving automatically. You cannot block the frequent events yourself, or choose which events to block.

If you want to find out if an event is blocked, you can view the notification list or you can check if this event is present in the **Blocking frequent events** section of the Administration Server properties. If the event is blocked, you can do the following:

- If you want to prevent overwriting the database, you can continue blocking such type of events from receiving.
- If you want, for example, to find the reason of sending the frequent events to the Administration Server, you can unblock frequent events and continue receiving the events of this type anyway.
- If you want to continue receiving the frequent events until they become blocked again, you can remove from blocking the frequent events.

Managing frequent events blocking

Administration Server blocks the automatic receiving of frequent events, but you can unblock and continue to receive frequent events. You can also block receiving frequent events that you unblocked before.

To manage frequent events blocking:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Blocking frequent events** section.

3. In the **Blocking frequent events** section:

   - If you want to unblock the receiving of frequent events:

     a. Select the frequent events you want to unblock, and then click the **Exclude** button.

     b. Click the **Save** button.

   - If you want to block receiving frequent events:

     a. Select the frequent events you want to block, and then click the **Block** button.

     b. Click the **Save** button.

Administration Server receives the unblocked frequent events and does not receive the blocked frequent events.

Removing blocking of frequent events

You can remove blocking for frequent events and start receiving them until Administration Server blocks these frequent events again.

To remove blocking for frequent events:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Blocking frequent events** section.

3. In the **Blocking frequent events** section, select the frequent event types for which you want to remove blocking.

4. Click the **Remove from blocking** button.

The frequent event is removed from the list of frequent events. Administration Server will receive events of this type.

Event processing and storage on the Administration Server

Information about events that occur during the operation of the application and managed devices is saved in the Administration Server database. Each event is attributed to a certain type and level of severity (*Critical event*, *Functional failure*, *Warning*, or *Info*). Depending on the conditions under which an event occurred, the application can assign different levels of severity to events of the same type.

You can view types and levels of severity assigned to events in the **Event configuration** section of the Administration Server properties window. In the **Event configuration** section, you can also configure processing of every event by the Administration Server:

- Registration of events on the Administration Server and in event logs of the operating system on a device and on the Administration Server.
- Method used for notifying the administrator of an event (for example, an SMS or email message).

In the **Events repository** section of the Administration Server properties window, you can edit the settings of events storage in the Administration Server database by limiting the number of event records and record storage term. When you specify the maximum number of events, the application calculates an approximate amount of storage space required for the specified number. You can use this approximate calculation to evaluate whether you have enough free space on the disk to avoid database overflow. The default capacity of the Administration Server database is 400,000 events. The maximum recommended capacity of the database is 45 million events.

The application checks the database every 10 minutes. If the number of events reaches the specified maximum value plus 10,000, the application deletes the oldest events so that only the specified maximum number of events remains.

When the Administration Server deletes old events, it cannot save new events to the database. During this period, information about events that were rejected is written to the operating system log. The new events are queued and then saved to the database after the deletion operation is complete. By default, the event queue is limited to 20,000 events. You can customize the queue limit by editing the KLEVP_MAX_POSTPONED_CNT flag value.

Notifications and device statuses

This section contains information on how to view notifications, configure notification delivery, use device statuses, and enable changing device statuses.

Using notifications

Notifications alert you about events and help you to speed up your responses to these events by performing recommended actions or actions you consider as appropriate.

Depending on the notification method chosen, the following types of notifications are available:

- Onscreen notifications
- Notifications by SMS

- Notifications by email
- Notifications by executable file or script

Onscreen notifications

Onscreen notifications alert you to events grouped by importance levels (*Critical*, *Warning*, and *Informational*).

Onscreen notification can have one of two statuses:

- *Reviewed*. It means you have performed recommended action for the notification, or you have assigned this status for the notification manually.
- *Not Reviewed*. It means you have not performed recommended action for the notification, or you have not assigned this status for the notification manually.

By default, the list of notifications include notifications in the *Not Reviewed* status.

You can monitor your organization's network viewing onscreen notifications and responding to them in a real time.

Notifications by email, by SMS, and by executable file or a script

Open Single Management Platform provides the capability to monitor your organization's network by sending notifications about any event that you consider important. For any event, you can configure notifications by email, by SMS, or by running an executable file or a script.

Upon receiving notifications by email or by SMS, you can decide on your response to an event. This response should be the most appropriate for your organization's network. By running an executable file or a script, you predefine a response to an event. You can also consider running an executable file or a script as a primary response to an event. After the executable file runs, you can take other steps to respond to the event.

Viewing onscreen notifications

You can view notifications onscreen in three ways:

- In the **Monitoring** → **Notifications** section. Here you can view notifications relating to predefined categories.
- In a separate window that can be opened no matter which section you are using at the moment. In this case, you can mark notifications as reviewed.
- In the **Notifications by selected severity level** widget on the **Monitoring** → **Dashboard** section. In the widget, you can view only notifications of events that are at the *Critical* and *Warning* importance levels.

You can perform actions, for example, you can response to an event.

To view notifications from predefined categories:

1. In the main menu, go to **Monitoring** → **Notifications**.

   The **All notifications** category is selected in the left pane, and in the right pane, all the notifications are displayed.

2. In the left pane, select one of the categories:

   - **Deployment**
   - **Devices**
   - **Protection**
   - **Updates** (this includes notifications about Kaspersky applications available for download, notifications about anti-virus database updates that have been downloaded, as well as notifications about the devices (if any) that require to be restarted after Network Agent update)
   - **Exploit Prevention**
   - **Administration Server** (this includes events concerning only Administration Server)
   - **Useful links** (this includes links to Kaspersky resources, for example, Kaspersky Technical Support, Kaspersky forum, license renewal page, or the Kaspersky IT Encyclopedia)

A list of notifications of the selected category is displayed. The list contains the following:

- Icon related to the topic of the notification: deployment ( ), protection ( ), updates ( ), device management ( ), Exploit Prevention ( ), Administration Server ( ).
- Notification importance level. Notifications of the following importance levels are displayed: **Critical notifications** ( ), **Warning notifications** ( ), **Info notifications**. Notifications in the list are grouped by importance levels.
- **Notification**. This contains a description of the notification.
- **Action**. This contains a link to a quick action that we recommend you perform. For example, by clicking this link, you can proceed to the repository and install security applications on devices, or view a list of devices or a list of events. After you perform the recommended action for the notification, this notification is assigned the *Reviewed* status.
- **Status registered**. This contains the number of days or hours that have passed from the moment when the notification was registered on the Administration Server.

To view onscreen notifications in a separate window by importance level:

1. In the upper-right corner of OSMP Console, click the flag icon ( ).

   If the flag icon has a red dot, there are notifications that have not been reviewed.

   A window opens listing the notifications. By default, the **All notifications** tab is selected and the notifications are grouped by importance level: *Critical*, *Warning*, and *Info*.

2. Select the **System** tab.

The list of *Critical* ( ) and *Warning* ( ) importance levels notifications is displayed. The notification list includes the following:

- Color marker. Critical notifications are marked in red. Warning notifications are marked in yellow.
- Icon indicating the topic of the notification: deployment ( ), protection ( ), updates ( ), device management ( ), Exploit Prevention ( ), Administration Server ( ).
- Description of the notification.
- Flag icon. The flag icon is gray if notifications have been assigned the *Not Reviewed* status. When you select the gray flag icon and assign the *Reviewed* status to a notification, the icon changes color to white.
- Link to the recommended action. When you perform the recommended action after clicking the link, the notification gets the *Reviewed* status.
- Number of days that have passed since the date when the notification was registered on the Administration Server.

3. Select the **More** tab.

   The list of *Info* importance level notifications is displayed.

   The organization of the list is the same as for the list on the **System** tab (see the description above). The only difference is the absence of a color marker.

You can filter notifications by the date interval when they were registered on Administration Server. Use the **Show filter** check box to manage the filter.

To view onscreen notifications in the widget:

1. In the **Dashboard** section, select **Add or restore widget**.
2. In the window that opens, click the **Other** category, select the **Notifications by selected severity level** widget, and click Add.

   The widget now appears on the **Dashboard** tab. By default, the notifications of *Critical* importance level are displayed on the widget.

   You can click the **Settings** button on the widget and change the widget settings to view notifications of the *Warning* importance level. Or, you can add another widget: **Notifications by selected severity level**, with a *Warning* importance level.

   The list of notifications on the widget is limited by its size and includes two notifications. These two notifications relate to the latest events.

The notification list in the widget includes the following:

- Icon related to the topic of the notification: deployment ( ), protection ( ), updates ( ), device management ( ), Exploit Prevention ( ), Administration Server ( ).
- Description of the notification with a link to the recommended action. When you perform a recommended action after clicking the link, the notification gets the *Reviewed* status.
- Number of days or number of hours that have passed since the date when the notification was registered on the Administration Server.
- Link to other notifications. Upon clicking this link, you are transferred to the view of notifications in the **Notifications** section of the **Monitoring** section.

About device statuses

Open Single Management Platform assigns a status to each managed device. The particular status depends on whether the conditions defined by the user are met. In some cases, when assigning a status to a device, Open Single Management Platform takes into consideration the device's visibility flag on the network (see the table below). If Open Single Management Platform does not find a device on the network within two hours, the visibility flag of the device is set to *Not Visible*.

The statuses are the following:

- *Critical* or *Critical/Visible*
- *Warning* or *Warning/Visible*
- *OK* or *OK/Visible*

The table below lists the default conditions that must be met to assign the *Critical* or *Warning* status to a device, with all possible values.

Conditions for assigning a status to a device

Condition	Condition description	Available values
Security application is not installed	Network Agent is installed on the device, but a security application is not installed.	• Toggle button is on. • Toggle button is off.
Too many viruses detected	Some viruses have been found on the device by a task for virus detection, for example, the Malware scan task, and the number of viruses found exceeds the specified value.	More than 0.
Real-time protection level differs from the level set by the Administrator	The device is visible on the network, but the real-time protection level differs from the level set (in the condition) by the administrator for the device status.	• Stopped. • Paused. • Running.
Malware scan has not been performed in a long time	The device is visible on the network and a security application is installed on the device, but neither the *Malware scan* task nor a local scan task has been run within the specified time interval. The condition is applicable only to devices that were added to the Administration Server database 7 days ago or earlier.	More than 1 day.
Databases are outdated	The device is visible on the network and a security application is installed on the device, but the anti-virus databases have not been updated on this device within the specified time interval. The condition is applicable only to devices that were added to the Administration Server database 1 day ago or earlier.	More than 1 day.
Not connected in a long time	Network Agent is installed on the device, but the device has not connected to an Administration Server within the specified time interval, because the device was turned off.	More than 1 day.
Active threats are detected	The number of unprocessed objects in the **Active threats** folder exceeds the specified value.	More than 0 items.
Restart is required	The device is visible on the network, but an application requires the device restart longer than the specified time interval and for one of the selected reasons.	More than 0 minutes.
Incompatible applications are installed	The device is visible on the network, but software inventory performed through Network Agent has detected incompatible applications installed on the device.	• Toggle button is off. • Toggle button is on.
Software vulnerabilities have been detected	The device is visible on the network and Network Agent is installed on the device, but the *Find vulnerabilities and required updates* task has detected vulnerabilities with the specified severity level in applications installed on the device.	• Critical. • High. • Medium. • Ignore if the vulnerability cannot be fixed. • Ignore if an update is assigned for installation.
License expired	The device is visible on the network, but the license has expired.	• Toggle button is off. • Toggle button is on.

Condition	Condition description	Available values
License expires soon	The device is visible on the network, but the license will expire on the device in less than the specified number of days.	Less than 0 days.
Check for Windows Update updates has not been performed in a long time	The device is visible on the network, but the *Find vulnerabilities and required updates* task has not been run within the specified time interval.	More than 1 day.
Invalid encryption status	Network Agent is installed on the device, but the device encryption result is equal to the specified value.	• Does not comply with the policy due to the user's refusal (for external devices only). • Does not comply with the policy due to an error. • Restart is required when applying the policy. • No encryption policy is specified. • Not supported. • When applying the policy.
Mobile device settings do not comply with the policy	The mobile device settings are other than the settings that were specified in the Kaspersky Endpoint Security for Android policy during the check of compliance rules.	• Toggle button is off. • Toggle button is on.
Unprocessed security issues detected	Some unprocessed security issues have been found on the device. Security issues can be created either automatically, through managed Kaspersky applications installed on the client device, or manually by the administrator.	• Toggle button is off. • Toggle button is on.
Device status defined by application	The status of the device is defined by the managed application.	• Toggle button is off. • Toggle button is on.
Device is out of disk space	Free disk space on the device is less than the specified value, or the device could not be synchronized with the Administration Server. The *Critical* or *Warning* status is changed to the *OK* status when the device is successfully synchronized with the Administration Server and free space on the device is greater than or equal to the specified value.	More than 0 MB.
Device has become unmanaged	During device discovery, the device was recognized as visible on the network, but more than three attempts to synchronize with the Administration Server failed.	• Toggle button is off. • Toggle button is on.
Protection is disabled	The device is visible on the network, but the security application on the device has been disabled for longer than the specified time interval. In this case, the state of the security application is *stopped* or *failure*, and differs from the following: *starting*, *running*, or *suspended*.	More than 0 minutes.
Security application is not running	The device is visible on the network and a security application is installed on the device but is not running.	• Toggle button is off. • Toggle button is on.

Open Single Management Platform allows you to set up automatic switching of the status of a device in an administration group when specified conditions are met. When the specified conditions are met, the client device is assigned one of the following statuses: *Critical* or *Warning*. When the specified conditions are not met, the client device is assigned the *OK* status.

Different statuses may correspond to different values of one condition. For example, by default, if the **Databases are outdated** condition has the **More than 3 days** value, the client device is assigned the *Warning* status; if the value is **More than 7 days**, the *Critical* status is assigned.

When Open Single Management Platform assigns a status to a device, for some conditions (see the Condition description column in the table above) the visibility flag is taken into consideration. For example, if a managed device was assigned the *Critical* status because the Databases are outdated condition was met, and later the visibility flag was set for the device, then the device is assigned the *OK* status.

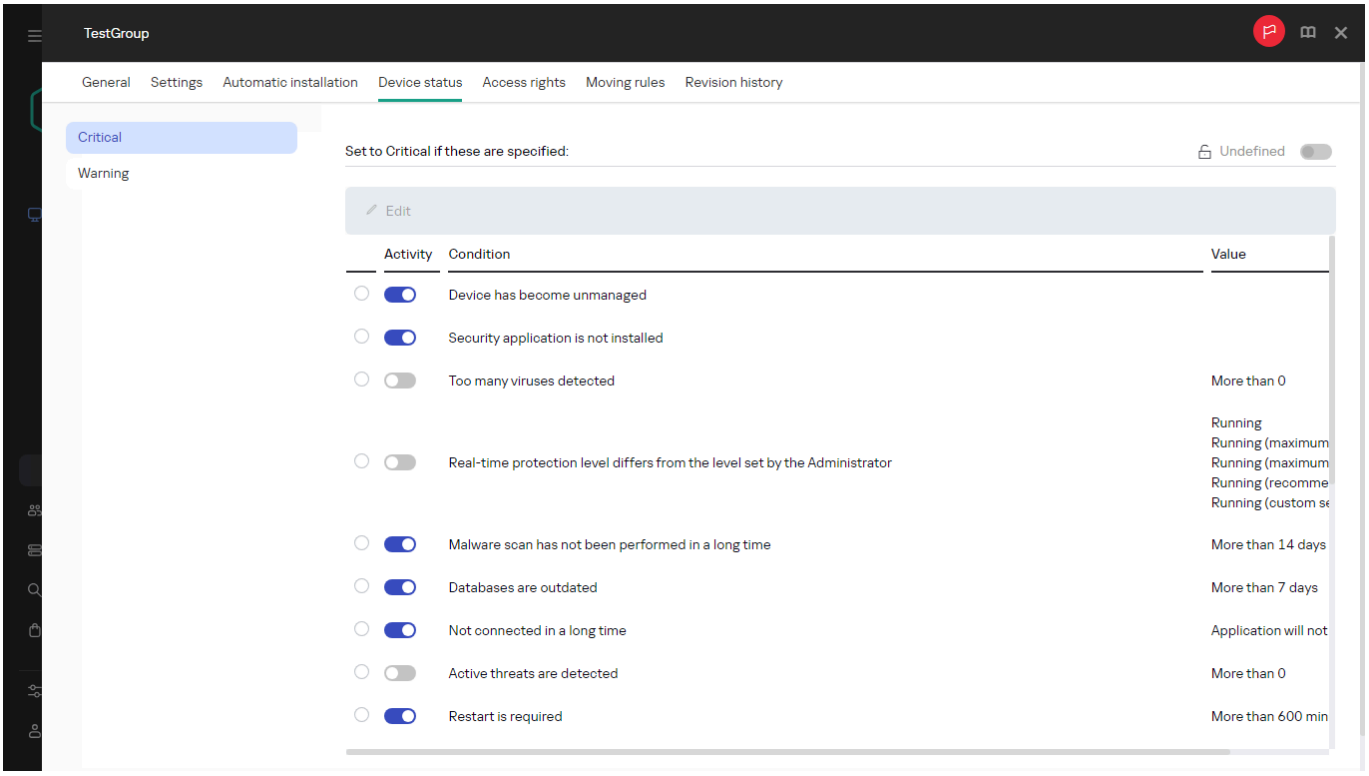
Configuring the switching of device statuses

You can change conditions to assign the *Critical* or *Warning* status to a device.

To enable changing the device status to Critical:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.
2. Go to the properties of the administration group for which you want to change the device statuses in one of the following ways:
   - Click the name of the administration group.
   - Click the **Show hierarchy** button at the top of the page. In the navigation pane that opens, hover the mouse cursor over the name of the required group, and then click the settings icon (⚙).

   The administration group properties window opens.
3. In the properties window, select the **Device status** tab.
4. In the left pane, select **Critical**.
5. In the right pane, in the **Set to Critical if the following settings are specified** section, enable the condition to switch a device to the *Critical* status.



> You can change only settings that are not locked in the parent policy.

6. Select the radio button next to the condition in the list.
7. In the upper-left corner of the list, click the **Edit** button.

8. Set the required value for the selected condition.

   Values cannot be set for every condition.

9. Click **OK**.

When specified conditions are met, the managed device is assigned the *Critical* status.

To enable changing the device status to Warning:

1. In the main menu, go to **Asset management** → **Hierarchy of groups**.
2. In the list of groups that opens, click the link with the name of a group for which you want to change switching the device statuses.
3. In the properties window that opens, select the **Device status** tab.
4. In the left pane, select **Warning**.
5. In the right pane, in the **Set to Warning if the following settings are specified** section, enable the condition to switch a device to the *Warning* status.

   You can change only settings that are not locked in the parent policy.

6. Select the radio button next to the condition in the list.
7. In the upper-left corner of the list, click the **Edit** button.
8. Set the required value for the selected condition.

   Values cannot be set for every condition.

9. Click **OK**.

When specified conditions are met, the managed device is assigned the *Warning* status.

Configuring notification delivery

You can configure notification about events occurring in Open Single Management Platform. Depending on the notification method chosen, the following types of notifications are available:

- Email—When an event occurs, Open Single Management Platform sends a notification to the email addresses specified.
- SMS—When an event occurs, Open Single Management Platform sends a notification to the phone numbers specified.
- Executable file—When an event occurs, the executable file is run on the Administration Server.

To configure notification delivery of events occurring in Open Single Management Platform:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens with the **General** tab selected.

2. Click the **Notification** section, and in the right pane select the tab for the notification method you want:

- **Email**

The **Email** tab allows you to configure event notification by email.

In the **SMTP servers** field, specify mail server addresses, separating them with semicolons. You can use the following values:

- IPv4 or IPv6 address
- DNS name of the SMTP server

In the **SMTP server port** field, specify the number of an SMTP server communication port. The default port number is 25.

If you enable the **Use DNS MX lookup** option, you can use several MX records of the IP addresses for the same DNS name of the SMTP server. The same DNS name may have several MX records with different values of priority of receiving email messages. Administration Server attempts to send email notifications to the SMTP server in ascending order of MX records priority.

> If you enable the **Use DNS MX lookup** option and do not enable usage of TLS settings, we recommend that you use the DNSSEC settings on your server device as an additional measure of protection for sending email notifications.

If you enable the **Use ESMTP authentication** option, you can specify the ESMTP authentication settings in the **User name** and **Password** fields. By default, the option is disabled, and the ESMTP authentication settings are not available.

You can specify TLS settings of connection with an SMTP server:

- **Do not use TLS**

You can select this option if you want to disable encryption of email messages.

- **Use TLS if supported by the SMTP server**

You can select this option if you want to use a TLS connection to an SMTP server. If the SMTP server does not support TLS, Administration Server connects the SMTP server without using TLS.

- **Always use TLS, check server certificate validity**

You can select this option if you want to use TLS authentication settings. If the SMTP server does not support TLS, Administration Server cannot connect the SMTP server.

> We recommend that you use this option for better protection of the connection with an SMTP server. If you select this option, you can set authentication settings for a TLS connection.

If you select **Always use TLS, check server certificate validity** value, you can specify a certificate for authentication of the SMTP server and choose whether you want to enable communication through any version of TLS or only through TLS 1.2 or later versions. Also, you can specify a certificate for client authentication on the SMTP server.

You can specify certificates for a TLS connection by clicking the **Specify certificates** link:

- Browse for an SMTP server certificate file:

You can receive a file with the list of certificates from a trusted certification authority and upload the file to Administration Server. Open Single Management Platform checks whether the certificate of an SMTP server is also signed by a trusted certification authority. Open Single Management Platform cannot connect to an SMTP server if the certificate of the SMTP server is not received from a trusted certification authority.

- Browse for a client certificate file:

You can use a certificate that you received from any source, for example, from any trusted certification authority. You must specify the certificate and its private key by using one of the following certificate types:

- X-509 certificate:

You must specify a file with the certificate and a file with the private key. Both files do not depend on each other and the order of loading of the files is not significant. When both files are loaded, you must specify the password for decoding the private key. The password can have an empty value if the private key is not encoded.

- pkcs12 container:

You must upload a single file that contains the certificate and its private key. When the file is loaded, you must then specify the password for decoding the private key. The password can have an empty value if the private key is not encoded.

Clicking the **Send test message** button allows you to check whether you configured notifications properly: the application sends a test notification to the email addresses that you specified.

In the **Recipients (email addresses)** field, specify the email addresses to which the application will send notifications. You can specify multiple addresses in this field, by separating them with semicolons.

In the **Subject** field, specify the email subject. You can leave this field empty.

In the **Subject template** drop-down list, select the template for your subject. A variable determined by the selected template is placed automatically in the **Subject** field. You can construct an email subject selecting several subject templates.

In the **Sender email address: If this setting is not specified, the recipient address will be used instead. Warning: We do not recommend using a fictitious email address** field, specify the sender email address. If you leave this field empty, by default, the recipient address is used. It is not recommended to use fictitious email addresses.

The **Notification message** field contains standard text with information about the event that the application sends when an event occurs. This text includes substitute parameters, such as event name, device name, and domain name. You can edit the message text by adding other substitute parameters with more relevant details about the event.

> If the notification text contains a percent sign (%), you have to type it twice in a row to allow message sending. For example, "CPU load is 100%%".

Clicking the **Configure numeric limit of notifications** link allows you to specify the maximum number of notifications that the application can send during the specified time interval.

- SMS

The **SMS** tab allows you to configure the transmission of SMS notifications about various events to a cell phone. SMS messages are sent through a mail gateway.

In the **SMTP servers** field, specify mail server addresses, separating them with semicolons. You can use the following values:

- IPv4 or IPv6 address
- DNS name of the SMTP server

In the **SMTP server port** field, specify the number of an SMTP server communication port. The default port number is 25.

If the **Use ESMTP authentication** option is enabled, you can specify the ESMTP authentication settings in the **User name** and **Password** fields. By default, the option is disabled, and the ESMTP authentication settings are not available.

You can specify TLS settings of connection with an SMTP server:

- **Do not use TLS**

You can select this option if you want to disable encryption of email messages.

- **Use TLS if supported by the SMTP server**

You can select this option if you want to use a TLS connection to an SMTP server. If the SMTP server does not support TLS, Administration Server connects the SMTP server without using TLS.

- **Always use TLS, check server certificate validity**

You can select this option if you want to use TLS authentication settings. If the SMTP server does not support TLS, Administration Server cannot connect the SMTP server.

> We recommend that you use this option for better protection of the connection with an SMTP server. If you select this option, you can set authentication settings for a TLS connection.

If you select **Always use TLS, check server certificate validity** value, you can specify a certificate for authentication of the SMTP server and choose whether you want to enable communication through any version of TLS or only through TLS 1.2 or later versions. Also, you can specify a certificate for client authentication on the SMTP server.

You can specify SMTP server certificate file by clicking the **Specify certificates** link. You can receive a file with the list of certificates from a trusted certification authority and upload the file to Administration Server. Open Single Management Platform checks whether the certificate of an SMTP server is also signed by a trusted certification authority. Open Single Management Platform cannot connect to an SMTP server if the certificate of the SMTP server is not received from a trusted certification authority.

In the **Recipients (email addresses)** field, specify the email addresses to which the application will send notifications. You can specify multiple addresses in this field, by separating them with semicolons. The notifications will be delivered to the phone numbers associated with the specified email addresses.

In the **Subject** field, specify the email subject.

In the **Subject template** drop-down list, select the template for your subject. A variable according to the selected template is put in the **Subject** field. You can construct an email subject selecting several subject templates.

In the **Sender email address: If this setting is not specified, the recipient address will be used instead. Warning: We do not recommend using a fictitious email address** field, specify the sender email address. If you leave this field empty, by default, the recipient address is used. It is not recommended to use fictitious email addresses.

In the **Phone numbers of SMS message recipients** field, specify the cell phone numbers of the SMS notification recipients.

In the **Notification message** field, specify a text with information about the event that the application sends when an event occurs. This text can include substitute parameters, such as event name, device name, and domain name.

If the notification text contains a percent sign (%), you have to type it twice in a row to allow message sending. For example, "CPU load is 100%%".

Click the **Send test message** to check whether you configured notifications properly: the application sends a test notification to the recipient that you specified.

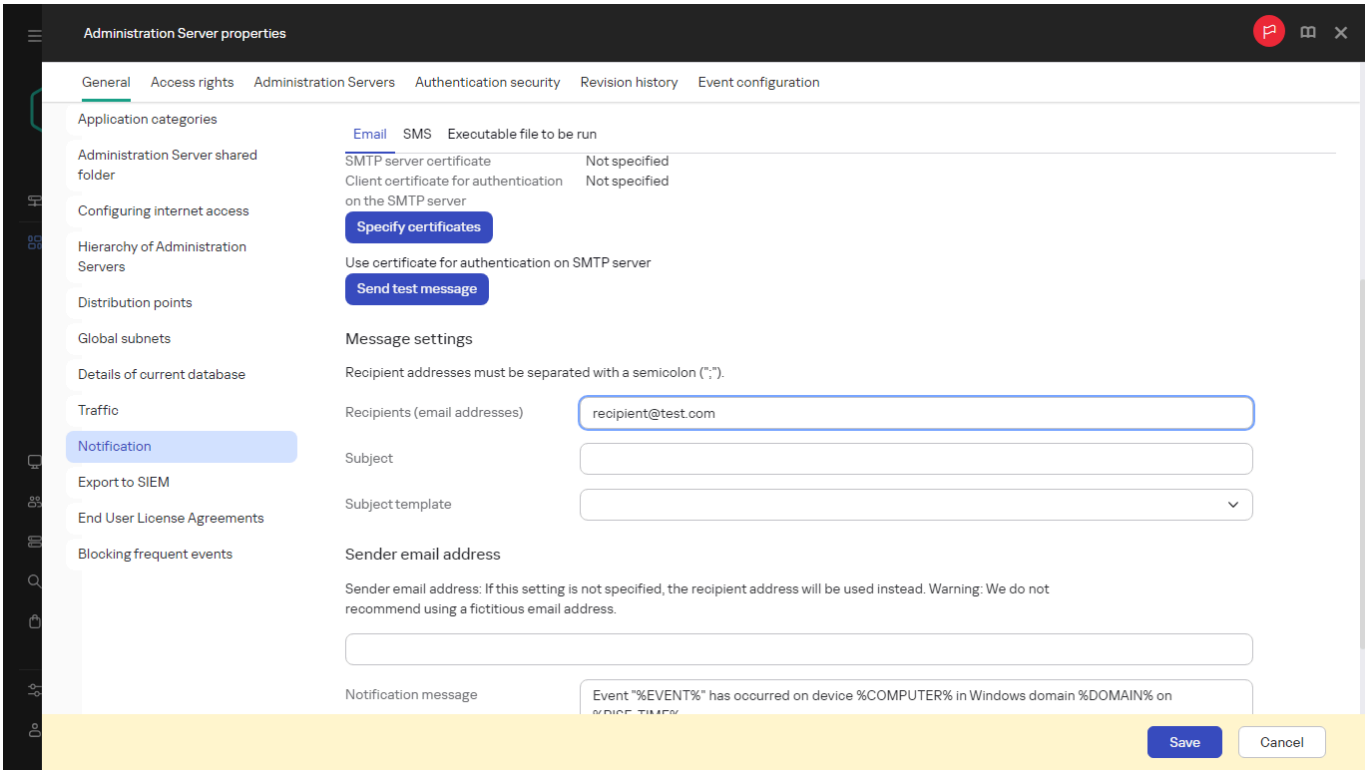
Click the **Configure numeric limit of notifications** link to specify the maximum number of notifications that the application can send during the specified time interval.

- **Executable file to be run**

If this notification method is selected, in the entry field you can specify the application that will start when an event occurs.

In the **Executable file to be run on the Administration Server when an event occurs** field, specify the folder and the name of the file to be run. Before specifying the file, prepare the file and specify the placeholders that define the event details to be sent in the notification message. The folder and the file that you specify must be located on the Administration Server.

Clicking the **Configure numeric limit of notifications** link allows you to specify the maximum number of notifications that the application can send during the specified time interval.



Selecting the notification method

3. On the tab, define the notification settings.

4. Click the **OK** button to close the Administration Server properties window.

The saved notification delivery settings are applied to all events that occur in Open Single Management Platform.

You can override notification delivery settings for certain events in the **Event configuration** section of the Administration Server settings, of a policy's settings, or of an application's settings.

Testing notifications

To check whether event notifications are sent, the application uses the notification of the EICAR test virus detection on client devices.

To verify sending of event notifications:

1. Stop the real-time file system protection task on a client device and copy the EICAR test virus to that client device. Then, re-enable real-time protection of the file system.

2. Run a scan task for client devices in an administration group or for specific devices, including one with the EICAR test virus.

   If the scan task is configured correctly, the test virus will be detected. If notifications are configured correctly, you are notified that a virus has been detected.

To open a record of the test virus detection:

1. In the main menu, go to **Monitoring** → **Event selections**.

2. Click the **Recent events** selection name.

   In the window that opens, the notification about the test virus is displayed.

> The EICAR test virus contains no code that can do harm to your device. However, most manufacturers' security applications identify this file as a virus. You can download the test virus from the official EICAR website.

Event notifications displayed by running an executable file

Open Single Management Platform can notify the administrator about events on client devices by running an executable file. The executable file must contain another executable file with Environment variables of the event to be relayed to the administrator (see the table below).

Environment variables for describing an event

Environment variable	Environment variable description
$SEVERITY	Event severity. Possible values: • **Info** • **Warning** • **Error** • **Critical**
$COMPUTER	Name of the device where the event occurred. Maximum length of the device name is 256 characters.
$DOMAIN	Domain name of the device where the event occurred.
$EVENT	Name of the event type. Maximum length of the event type name is 50 characters.

Environment variable	Environment variable description
$DESCR	Event description. Maximum length of the description is 1000 characters.
$RISE_TIME	Event creation time.
$KLCSAK_EVENT_TASK_DISPLAY_NAME	Task name. Maximum length of the task name is 100 characters.
$KL_PRODUCT	Application name.
$KL_VERSION	Application version number.
$KLCSAK_EVENT_SEVERITY_NUM	Event severity number. Possible values: • 1—**Info** • 2—**Warning** • 3—**Error** • 4—**Critical**
$HOST_IP	IP address of the device where the event occurred.
$HOST_CONN_IP	Connection IP address of the device where the event occurred.

Example:
Event notifications are sent by an executable file (such as script1.bat) inside which another executable file (such as script2.bat) with the $COMPUTER environment variable is launched. When an event occurs, the script1.bat file is run on the administrator's device, which, in turn, runs the script2.bat file with the $COMPUTER environment variable. The administrator then receives the name of the device where the event occurred.

Monitoring event sources

This section provides information about monitoring event sources.

Source status

In KUMA, you can monitor the state of the sources of data received by collectors. There can be multiple sources of events on one server, and data from multiple sources can be received by one collector.

You can configure automatic identification of event sources using one of the following sets of fields:

- Custom set of fields. You can specify from 1 to 9 fields in the order you want. TenantID does not need to be specified separately, it is determined automatically.
- **Apply default mapping** — DeviceProduct, DeviceHostName, DeviceAddress, DeviceProcessName. The field order cannot be changed.

  Sources are identified if the following fields in events are not empty: the DeviceProduct field, the DeviceAddress and/or DeviceHostname field, and the TenantID field (you do not need to specify the TenantID field, it is determined automatically). The DeviceProcessName field can be empty. If the DeviceProcessName field is not empty, and the other required fields are filled, a new source is identified.

  Identification of event sources depending on non-empty event fields

DeviceProduct	DeviceHostName	DeviceAddress	DeviceProcessName	TenantID (determined automatically)	
+	+			+	Source 1 identified
+		+		+	Source 2 identified
+	+	+		+	Source 3 identified

DeviceProduct	DeviceHostName	DeviceAddress	DeviceProcessName	TenantID (determined automatically)	
+	+		+	+	Source 4 identified
+		+	+	+	Source 5 identified
+	+	+	+	+	Source 6 identified
	+	+		+	Source not identified
	+		+	+	Source not identified
		+	+	+	Source not identified
+			+	+	Source not identified

Only one set of fields is applied for the entire installation. When upgrading to a new KUMA version, the default set of fields is applied. Only a user with the General Administrator role can configure the set of fields for identifying an event source. After you save changes to the set of fields, previously identified event sources are deleted from the KUMA Console and from the database. If necessary, you can revert to using a set of fields to determine default event sources. For the edited settings to take effect and KUMA to begin identifying sources based on the new settings, you must restart the collectors.

To identify event sources:

1. In the main menu, go to **Monitoring** → **Source status**.

2. This opens the **Source status** window; in that window, click the **Settings** button.

3. This opens the **Settings of event source detection** window; in that window, in the **Grouping fields for source detection** drop-down list, select the event fields by which you want to identify event sources.

   You can specify from 1 to 9 fields in the order you want. In a custom configuration, KUMA identifies sources in which the TenantID field is filled (you do not need to specify this field separately, it is determined automatically) and at least one field from the **Identical fields for source identification** is filled. For numeric fields, 0 is considered an empty value. If a single numeric field is selected for source identification, and the value of the numeric field is 0, the source is not detected.

   After you save the modified set of fields, an audit event is created and all previously identified sources are deleted from the KUMA Console and from the database; assigned policies are disabled.

4. If you want to go back to the list of fields for identifying the default event source, click **Apply default mapping**. The default field order cannot be changed. If you manually specify the fields in the wrong order, an error is displayed and the save settings button becomes unavailable. The correct default sequence of fields is DeviceProduct, DeviceHostName, DeviceAddress, DeviceProcessName. Minimum configuration for identifying event sources using the default set of events: non-empty values in the DeviceProduct field, the DeviceAddress and/or DeviceHostName field, and the TenantID field (TenantID is determined automatically).

5. Click **Save**.

6. Restart the collectors to apply the changes and begin identifying event sources by the specified list of fields.

Source identification is configured.

To view events that are associated with an event source:

1. In the main menu, go to **Monitoring** → **Source status**.

   This opens the **Source status** window.

2. On the **Event sources** tab, select your event source in the list, and in the **Name** column, expand the menu for the selected event source, click the **Events for <number> days** button.

   KUMA takes you to the **Events** section, where you can view a list of events for the selected source over the last 5 minutes. Values of fields configured in the event source identification settings are automatically specified in the query. If necessary, in the **Events** section, you can change the time period in the query and click **Run query** again to view the queried data for the specified time period.

Limitations

1. In a configuration with the default field set, KUMA registers the event source only if the raw event contains the DeviceProduct field and the DeviceAddress and/or DeviceHostName fields.

   If the raw event does not contain the DeviceProduct field and the DeviceAddress and/or DeviceHostName fields, you can:

   - Configure enrichment in the normalizer: on the **Enrichment** tab of the normalizer, select the **Event** data type, specify the **Source field** setting, and for the **Target field**, select the DeviceProduct + DeviceAddress and/or DeviceHostName and click OK.

   - Use an enrichment rule: select the **Event** data source type, specify the **Source field** setting, and as the **Target field**, select DeviceProduct + DeviceAddress and/or DeviceHostName, then click **Create**. The created enrichment rule must be linked to the collector at the Event enrichment step.

   KUMA will perform enrichment and register the event source.

2. If KUMA receives events with identical values of the fields that identify the source, KUMA registers different sources if the following conditions are satisfied:

   - The values of the required fields are identical, but different tenants are determined for the events.

   - The values of the required fields are identical, but one of the events has an optional DeviceProcessName field specified.

   - The values of the required fields are identical, but the data in these fields have different character case.

If you want KUMA to log such events under the same source, you can further configure the fields in the normalizer.

Lists of sources are generated in collectors, merged in the KUMA Core, and displayed in the program web interface under **Source status** on the **List of event sources** tab. Data is updated every minute.

The rate and number of incoming events serve as an important indicator of the state of the observed system. You can configure monitoring policies such that changes are tracked automatically and notifications are automatically created when indicators reach specific boundary values. Monitoring policies are displayed in the KUMA Console in the **Monitoring** → **Source status** section, on the **Monitoring policies** tab.

When monitoring policies are triggered, monitoring events are created and include data about the source of events.

List of event sources

Sources of events are displayed in the table under **Monitoring** → **Source status** → **Event sources**. One page can display up to 250 sources. You can sort the table by clicking the column heading of the relevant parameter and selecting **Ascending** or **Descending**.

You can use the **Search** field to search for event sources. The search is performed using regular expressions (RE2). You can also filter the table by the **Status** or **Monitoring policy** columns by clicking the heading of the relevant column and selecting the values that you want to display.

If necessary, you can configure the interval for updating data in the table. Available update periods: **1 minute**, **5 minutes**, **15 minutes**, **30 minutes**, **1 hour**. The default value is **No refresh**. You may need to configure the update period to track changes made to the list of sources.

Viewing information about event sources

On the **Event sources** tab, information about event sources is displayed in the following columns:

- **Status**—status of the event source:

  - Green—events are being received within the limits of the assigned monitoring policies.

  - Red—the frequency or number of incoming events go beyond the boundaries defined in at least one assigned monitoring policy.

  - Gray—monitoring policies have not been assigned to the source of events.

  If the status is red, an event of the Monitoring type generated. The monitoring event is generated in the tenant that owns the event source and is sent to the storage of the Main tenant (the storage must already be deployed in the Main tenant). If you have access to the tenant of the event source and do not have access to the Main tenant, you can still search for monitoring events in the storage of the Main tenant; the monitoring events of the tenants available to you will be displayed for you. You can also configure notifications to be sent to an arbitrary email address.

  The table can be filtered by status.

- **Name**—name of the event source. The name is generated automatically from the values of fields configured in the event source identification settings.

  You can rename an event source in the table of event sources by hovering over its name and clicking the pencil icon. The name can contain no more than 128 Unicode characters.

- **Host name or IP address**—name or IP address of the host from which the events originate if the DeviceHostName or DeviceAddress fields are specified in the event source identification settings.

- **Monitoring policy**—list of the monitoring policies assigned to the event source.

  If you want to filter the list of event sources by applied monitoring policies, click the name of this column and select one or more monitoring policies. If necessary, you can find policies in the list using the **Search** field.

  You can view information about all monitoring policies assigned to an event source by clicking the row of the source. This opens a window that displays the settings of monitoring policies, as well as the status of the source according to each policy. If several monitoring policies are assigned to the source, the red status in the table of sources in this window lets you identify the policy that was triggered. You can also see which policies are enabled and which are disabled, and when the disabled policies will be enabled again.

- **Stream**—frequency at which events are received from the event source. If only monitoring policies of the **byCount** type or monitoring policies of different types are assigned to the source, this value is displayed as the number of events. If only monitoring policies of the **byEPS** type are assigned to the source, or no policies are assigned, the value is displayed as the number of events per second.

- **Tenant**—the tenant that owns the events received from the event source.

Managing event sources

You can select one or more event sources by selecting the check boxes in the first column of the table. You can select multiple event sources at once for performing group operations by selecting the check box in the heading of the first column and selecting **Select all** or **Select all in page**. The **Select all in page** option applies only to event sources displayed in the list: if only 500 out of 1500 sources are displayed in the list, then group actions to download, enable or disable policies, or delete event sources are applied only to the selected 500 sources. If you want to perform an action on all sources in the table, select **Select all**.

If you select sources of events, the following buttons become available:

- The **Enable policy** button enables the monitoring policy for event sources. You must select policies in the displayed window to apply them.

- You can use the **Disable policy** to disable the monitoring policy for event sources. When disabling a policy, you must specify if want to disable the policy temporarily or forever.

- The **Update policy** button applies the monitoring policies that are enabled for the event sources, or change the monitoring policies that are already assigned. When a policy is updated, a task is started in the task manager.

  This button becomes available after you change the monitoring policies assigned to event sources.

- You can click the **Remove** button to remove event sources from the table. The statistics on this source will also be removed. If a collector continues to receive data from the source, the event source will re-appear in the table but its old statistics will not be taken into account.

  If you want to delete all event sources, but some time has passed since the table was last refreshed, sources added during this time may not be displayed in the table, but they will be deleted regardless.

  If you delete more than 100,000 event sources to which a filter or search was applied, only the first 100,000 event sources will be deleted. You can select all filtered event sources again and delete them, and then repeat this until you have deleted all event sources that you intended to delete. You can delete over 100,000 event sources if no filters or searches are applied to them by selecting sources using the **Select all** button.

- You can click **CSV** to download the data of the selected event sources to a CSV file.

  The `Stream` field is downloaded only if a monitoring policy has been assigned to the event source; in that case, the unit of measurement taken from the policy is specified in the downloaded file. If no policy is assigned an empty `Stream` field is expected behavior.

- You can click the **Chart** button to plot a chart of incoming events for the last seven days for the selected event sources. You can select up to five event sources
  .

Downloading event source information to a CSV file

You can download information about one or more event sources and the monitoring policies applied to them to a CSV file in UTF-8 encoding. If multiple monitoring policies are applied to a source, in the file for that source, each monitoring policy and its parameters starts on a new line. For each monitoring policy applied to a source, the following parameters are exported to the file: **Status**, **Name**, **Monitoring policy**, **Lower limit**, **Upper limit**, **Stream**, **Tenant**.

To download event source information to a CSV file:

1. In the main menu, go to **Monitoring** → **Source status** → **Event sources**, select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources.

In the lower left part of the table, you can find the number of selected sources and the total number of sources in the table. You can select up to 150,000 event sources.

You can select several event sources by clicking the check box in the heading of the first column and selecting one of the following options:

- **Select all** to select all event sources on all pages of the table. If you have used search to filter sources, this will select all sources that match the search query.
- **Select all in page** to select all event sources on the currently displayed page. If you have used search to filter sources, this will select all sources on the currently displayed page that match the search query.

2. Click the **CSV** button in the upper part of the table.

   Depending on the size of your browser window, the **CSV** button may be found in the additional menu that you can open by clicking on the icon with the three dots ⋮.

   A new event source export task is created in the task manager.

3. Go to the **Task manager** section and find the created task.

   When the file is ready, the **Status** column of the task displays the **Completed** status.

4. Click the task type name and select **Download** from the drop-down list.

The CSV file with event source information is downloaded in accordance with your browser settings. The default file name is event-source-list.csv.

Viewing the dynamics of incoming events

You can examine the dynamics of events received from a source over the last seven days, taking into account the applied monitoring policies, in one of the following ways:

- View the graph for an individual event source.
- Plot a chart based on graphs for several (up to five) sources.

To view the graph for a single event source, go to **Monitoring** → **Source status** → **Event sources**, and then click the arrow icon › in the row of the relevant event source. The graph of incoming events is displayed under the row of the source.

The data in the graph is displayed as follows:

- The data is displayed for the days on which the events were received. The maximum period is seven days.

  In the upper left corner above the graph, the number of days is displayed, and in the upper right corner, the data display period is displayed. You can click the **Events for <number> days** button to go to the **Events** section and view the list of events for the selected source.

- The X-axis represents days, and the Y-axis represents the frequency of events (EPS).
- The lines represent the average, maximum, and minimum number of events for every 15-minute period during the last seven days.

  If you want to view the number of events at a specific time, hover over a point on the graph. A tooltip is displayed with the average, maximum, and minimum event count at a specific date and time.

You can also plot a chart of incoming events based on graphs for several event sources, for example, if you need to compare the activity of event sources of the same type that should behave in a similar way, but in fact behave in different ways.

To plot a chart based on graphs for multiple event sources:

1. In the main menu, go to **Monitoring** → **Source status** → **Event sources**, then select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources.

   You can plot a chart for up to 5 event sources at the same time.

2. Click the **Chart** button in the upper part of the table.

   > Depending on the size of your browser window, the **Chart** button may be found in the additional menu that you can open by clicking on the icon with the three dots ⋮.

   The displayed **Chart** pane contains a chart of incoming events for all selected sources as well as a table that displays the current number of events, the maximum number of events, and the average number of events for each source, calculated based on the data from the chart. You can compare how the data for the selected sources relates to each other over time.

   The data in the chart is displayed as follows:

   - The data is displayed for the days on which the events were received. The maximum period is seven days.

     In the upper right corner above the chart, the data display period is displayed.

   - The X-axis represents days, and the Y-axis represents the frequency of events (EPS).

   - The lines in the chart represent the average number of incoming events from the selected event sources for every 15-minute interval during the last seven days.

   You can hover over the chart to view the average number of events for each source at a specific time.

3. If necessary, clear the check boxes in the table below the chart next to the event sources that you want to hide in the chart.

4. If you want to display the diagram in more detail, click the two arrows ↗↙ icon to open the panel in full screen mode and zoom in on the diagram.

Monitoring policies

The rate and number of incoming events serve as an important indicator of the state of the system. For example, you can detect when there are too many events, too few, or none at all. Monitoring policies are designed to detect such situations. In a policy, you can specify a lower threshold, an optional upper threshold, and the way the events are counted: by frequency or by total number.

The policy must be applied to the event source. You can apply one or more monitoring policies to a source. After applying the policy, you can monitor the status of the source on the **List of event sources** tab.

Policies for monitoring the sources of events are displayed in the table under **Monitoring** → **Source status** → **Monitoring policies**. You can sort the table by clicking the column header of the relevant setting. The maximum size of the policy list is not limited.

In the **Sources** column, you can click the **Show** button to view all event sources to which the policy is applied. When you click this button, you are taken to the **Event sources** tab, and the table of sources is filtered by the selected policy.

Algorithm of monitoring policies

Monitoring policies are applied to an event source in accordance with the following algorithm:

1. The event stream is counted at the collector.
2. The KUMA Core server gets information about the stream from the collectors every 15 seconds.
3. The obtained data is stored on the KUMA Core server in the Victoria Metrics time series database, and the data storage depth on the KUMA Core server is 15 days.
4. An inventory of event sources is taken once per minute.
5. The stream is counted separately for each event source in accordance with the following rules:
   - If a monitoring policy is applied to the event source, the displayed maximum number of events is calculated in accordance with the currently applied monitoring policies for the time interval specified in the policy.

     Depending on the policy type, the number of the event stream is counted as the number of events (for the **byCount** policy type) or as the number events per second (EPS, for the **byEPS** policy type). You can look up how the stream is counted for the applied policy in the **Stream** column on the **Event sources** tab.
   - If no monitoring policy is applied to the event source, the number for the event stream corresponds to the last value.
6. Once a minute, the application checks if any monitoring policies exist that must be applied to event sources or stopped according to the monitoring policy schedule.
7. Once a minute, the stream of events is checked for compliance with policy settings.

If the event stream from the source crosses the thresholds specified in the monitoring policy, information about this is recorded in the following way:

- A notification about a monitoring policy getting triggered is sent to the email addresses specified in the policy. For each policy, you can also configure a notification template.
- A stream monitoring informational event of type `5(Type=5)` is generated. The fields of the event are described in the table below.

Fields of the monitoring event

Event field name	Field value
ID	Unique ID of the event.
Timestamp	Event time.
Type	Type of the audit event. For the audit event, the value is `5` (monitoring).
Name	Name of the monitoring policy.
DeviceProduct	`KUMA`
DeviceCustomString1	The value from the `value` field in the notification. Displays the value of the metric for which the notification was sent.

The generated monitoring event is sent to the following resources:

- All storages of the Main tenant
- All correlators of the Main tenant
- All correlators of the tenant in which the event source is located

Adding a monitoring policy

To add a new monitoring policy:

1. In the main menu, go to **Monitoring** → **Source status** → **Monitoring policies**, then click **Add policy** and configure the monitoring policy in the displayed window:
2. In the **Policy name** field, enter a unique name for the policy you are creating. The name must contain 1 to 128 Unicode characters.

   We recommend choosing a name that reflects the configured schedule of the monitoring policy.
3. In the **Tenant** drop-down list, select the tenant that will own the policy. Your tenant selection determines the specific sources of events that can covered by the monitoring policy.
4. In the **Type** field, select one of the following monitoring policy types:
   - **by count**—by the number of events over a certain period of time.
   - **by EPS**—by the number of events per second (EPS) over a certain period of time. The average value over the entire period is calculated. You can additionally track spikes during specific periods.
5. In the **Count interval** field, specify the period during which the monitoring policy must take into account the data from the monitoring source. You can use the drop-down list on the right to select a value in minutes, hours, or days. The maximum value is 14 days.
6. If you selected the **by EPS** policy type, in the **Control interval, minutes** field, specify the control time interval (in minutes) within which the number of events must cross the threshold for the monitoring policy to trigger:
   - If, during this time period, all checks (performed once per minute) find that the stream is crossing the threshold, the monitoring policy is triggered.
   - If, during this time period, one of the checks (performed once per minute) finds that the stream is within the thresholds, the monitoring policy is not triggered, and the count of check results is reset.

   If you do not specify the frequency of measurement, the monitoring policy is triggered immediately after the stream is found to cross the threshold.
7. In the **Lower limit** and **Upper limit** fields, define the boundaries representing normal behavior. Deviations outside of these boundaries will trigger the monitoring policy, create an alert, and forward notifications.

   The **Lower limit** setting is required.

8. In the **Evaluation interval** field, specify the frequency with which the VMalert service will query VictoriaMetrics for policy data while the policy is being applied to the event source. You can use the drop-down list on the right to select a value in minutes, hours, or days. The default interval is 5 minutes.

   When specifying the evaluation interval, keep in mind the policy schedule. For example, if you configured the policy to be applied once every few hours, we do not recommend configuring a short interval and causing excessive load on VictoriaMetrics.

9. If necessary, in the **Send notifications** field, specify the email addresses to which notifications about the activation of the KUMA monitoring policy must be sent. To add an address, enter it in the field and press **`Enter`** or click **Add**. You can specify multiple email addresses.

   To forward notifications, you must configure a connection to the SMTP server.

10. In the **Notification template** drop-down list, select the template that you want to use for notifications. If necessary, click the **Create new** button to start creating a new notification template.

    By default, the basic notification template is selected. You can reset the template selection and switch to the base template by clicking the X icon.

11. In the **Schedule** section, configure how often you want to apply the monitoring policy to event sources. By default, the policy is applied every week, every day, from 00:00 to 23:59. To configure the monitoring policy schedule, do some of the following:

    - If you want to apply the monitoring policy weekly on specific days of the week:

      a. Enable the **Configure schedule by days of the week** toggle switch.

      b. In the **Days of the week** drop-down list, select the days of the week on which you want the policy to be applied to the source.

      If you want to clear the selection, click the X icon.

      c. In the **Time** field, specify the start and end time of the policy, with minute precision.

      The policy applicability interval is inclusive of its bounds; for example, if the end time is set to 23:59, the policy will be applied until 23:59:59.999. The default interval is 00:00 to 23:59. The start time must be earlier than the end time.

      d. If you want to add another period, click the **Add period** button and repeat steps 'b' and 'c'.

      You can add any number of periods.

    - If you want to apply the monitoring policy weekly on specific calendar dates:

      a. Enable the **Configure schedule by days of the month** toggle switch.

      b. Click the **Days of the month** field and use the calendar to select the dates on which you want to apply the policy to the source. You can select a period of several days or an individual day. The start date of the period must be earlier than the end date of the period.

      The dates are configured without a year value, so the policy will be applied annually on the specified days until you delete this period. If you want to clear the selection, click the X icon.

      c. In the **Time** field, specify the start and end time of the policy, with minute precision.

      The policy applicability interval is inclusive of its bounds; for example, if the end time is set to 23:59, the policy will be applied until 23:59:59.999. The default interval is 00:00 to 23:59. The start time must be earlier than the end time.

d. If you want to add another period, click the **Add period** button and repeat steps 'b' and 'c'.

You can add any number of periods.

If you applied a schedule by day of the week and by day of the month at the same time, the day-of-the-month policy is applied first.

12. Click **Add**.

The monitoring policy will be added.

Editing monitoring policies

The **Monitoring** → **Source status** → **Monitoring policies** section displays the added monitoring policies and their settings that you specified when creating the policy. You can click a policy to display a sidebar with all of its settings. If necessary, you can edit the policy settings in this sidebar.

If a monitoring policy is applied to an event source, if you edit certain policy settings, you may need to update the policy to apply the changes. Every 30 minutes, KUMA checks if any monitoring policies require updating, and if that is the case, it automatically runs a task to update those monitoring policies. You can also run the update task manually by clicking the **Update policy** button at the top of the table. One task updates all policies that need updating.

The **Update policy** button becomes active only if some monitoring policies need updating. Information about whether the policy needs updating is displayed in the table of monitoring policies in the **Policy update status** as one of the following statuses:

- **Update required** if one of the following monitoring policy settings was edited, but the changes have not been applied to event sources:
  - **Policy name**
  - **Type**
  - **Lower limit**
  - **Upper limit**
  - **Count interval**
  - **Control interval**
  - **Evaluation interval**
- **Updated** in any of the following cases:
  - After editing the policy, the task to apply the policy was started, and the changes were applied to event sources.
  - You have edited one of the following policy settings, which does not require starting the update task:
    - **Send notifications**
    - **Notification template**
    - **Schedule**

In this case, the edited policy settings are applied to event sources after a minute. Changes of the **Notification template** setting are applied instantly.

- The modified monitoring policy is not applied to event sources.

The date and time when the policy was last applied to event sources is displayed in the **Policy last applied** column.

> While the policy update task is running, the **Update policy** button is unavailable for all users. If another user has edited the settings of the policy that necessitate an update, the **Update policy** button becomes active for you only after you refresh the page or edit the policy or an event source.

Applying monitoring policies

To apply monitoring policies to event sources:

1. In the main menu, go to **Monitoring** → **Source status** → **Event sources**, then select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources. You can select several event sources by clicking the check box in the heading of the first column and selecting one of the following options:

   - **Select all** to select all event sources on all pages of the table. If you have used search to filter sources, this will select all sources that match the search query.

   - **Select all in page** to select all event sources that are loaded on the currently displayed page. If you have used search to filter sources, this will select all sources on the currently displayed page that match the search query.

   In the lower left part of the table, you can find the number of selected sources and the total number of sources in the table.

   After you select the event sources to which you want to apply the monitoring policy, the **Enable policy** button becomes available on the toolbar.

2. Click **Apply policy**.

3. This opens the **Apply policy** window; in that window, select one or more monitoring policies that you want to apply to the selected event sources. The table lists only monitoring policies that you can assign to the selected sources: policies that belong to the same tenant or to the Shared tenant, if you have access to it. If no shared policies exist for the selected event sources and you do not have access to the Shared tenant, the policy table is empty.

   To select all available policies, you can select the check box in the heading of the first column. You can also use context search by policy name or sort the policies by clicking the heading of the column by which you want to sort the table and selecting **Ascending** or **Descending**.

   > Search and sorting is not available for the **Sources**, **Schedule**, **Policy update status**, **Policy last applied** columns.

4. Click **Apply**.

5. In the table of sources, click **Update policy** to apply the changes to event sources.

   The monitoring policies are applied to the selected event sources; the status of these sources changes to green. The names of the policies applied to the sources are displayed in the **Monitoring policy** column. A message is also displayed indicating the number of sources to which the policies have been applied. If the monitoring policy is triggered for an event source, the new status of that source is displayed after you manually refresh the page or it is refreshed automatically. We recommend configuring an automatic data refresh period to keep track of changes in the list of sources.

If you have selected more than 100,000 event sources and applied one or more policies to them, these policies are applied only to the first 100,000 sources to which these policies have not yet been applied. If you need to apply policies to the remaining sources, you can do one of the following:

- Select all sources again and apply the policies to them.
- Filter the table of sources by any parameter so that the table displays less than 100,000 sources, then apply the policies to them.

Repeat the action until the policies have been applied to all the sources that you need.

Disabling monitoring policies

To disable monitoring policies for event sources:

1. In the main menu, go to **Monitoring** → **Source status** → **Event sources**, then select one or more event sources in the table by selecting the check boxes in the first column next to the relevant sources.

   In the lower left part of the table, you can find the number of selected sources and the total number of sources in the table. After you select the event sources to which monitoring policies are applied in the list, the **Disable policy** button becomes available on the toolbar.

   You can select several event sources by clicking the check box in the heading of the first column and selecting one of the following options:

   - **Select all** to select all event sources on all pages of the table. If you have used search to filter sources, this will select all sources that match the search query.
   - **Select all in page** to select all event sources that are loaded on the currently displayed page. If you have used search to filter sources, this will select all sources on the currently displayed page that match the search query.

2. Click **Disable policy**.

3. This opens the **Disable policy** window; in that window, select one or more monitoring policies that you want to disable for the selected event sources. The table lists all monitoring policies applied to at least one of the selected event sources.

   To select all available policies, you can select the check box in the heading of the first column. You can also use context search or sort the policies by clicking the heading of the column by which you want to sort the table and selecting **Ascending** or **Descending**.

   Search and sorting is not available for the **Sources**, **Schedule**, **Policy update status**, **Policy last applied** columns.

4. In the settings section above the policy table, do one of the following:

   - If you want to temporarily suspend the policies, select **For the specified time** and specify the time in minutes, hours, or days after which the selected policies will be reapplied to event sources. Maximum values:

     - For days: 30

     - For hours: 743

     - For minutes: 44579

   - If you want to permanently disable the selected policies for event sources, select **Until manually enabled**.

   The default selection is **For the specified time**, and the value is set to **5 minutes**.

5. Click **Disable**.

6. In the table of sources, click **Update policy** to apply the changes to event sources.

   The monitoring policies are disabled for selected event sources or suspended for the specified time. The status of these sources in the table changes to gray. A message is displayed indicating the number of sources for which the policies have been disabled.

If you have selected more than 100,000 event sources and disabled one or more policies for them, these policies are disabled only for the first 100,000 sources to which these policies are applied. If you need to disable policies for the remaining sources, you can do one of the following:

- Select all sources again and disable the policies for them.

- Filter the table of sources by any parameter so that the table displays less than 100,000 sources, then disable the policies for them.

Repeat the action until the policies have been disabled for all the sources that you need.

Adding a new monitoring policy based on an existing policy

To create a new monitoring policy based on an existing policy:

1. In the main menu, go to **Monitoring** → **Source status** → **Monitoring policies**, then select the monitoring policy that you want to base the new policy on.

   If necessary, you can find monitoring policies in the list using the **Search** field. The search will be carried out in the following columns: **Name**, **Tenant**, **Type**, **Schedule** (name of the day and time).

2. Click **Duplicate policy**.

3. This opens the **Add policy** window, in which you can edit policy settings.

   By default, `- copy` is appended to the name of the new policy. The rest of the settings are the same as in the policy that you are duplicating.

4. Click the **Add** button to create the new policy.

   The monitoring policy is created based on an existing policy.

Deleting monitoring policies

To delete a monitoring policy:

1. In the main menu, go to **Monitoring** → **Source status** → **Monitoring policies**, then select one or more monitoring policies that you want to delete.

   If necessary, you can find monitoring policies in the list using the **Search** field. The search will be carried out in the following columns: **Name**, **Tenant**, **Type**, **Schedule** (name of the day and time).

2. Click **Delete policy** and confirm the action.

The selected monitoring policies are deleted.

> You cannot remove predefined monitoring policies or policies that are assigned to data sources.

Kaspersky announcements

This section describes how to use, configure, and disable Kaspersky announcements.

About Kaspersky announcements

The Kaspersky announcements section (**Monitoring** → **Kaspersky announcements**) keeps you informed by providing information related to your version of Open Single Management Platform and the managed applications installed on the managed devices. Open Single Management Platform periodically updates the information in the section by removing outdated announcements and adding new information.

Open Single Management Platform shows only those Kaspersky announcements that relate to the currently connected Administration Server and the Kaspersky applications installed on the managed devices of this Administration Server. The announcements are shown individually for any type of Administration Server—primary, secondary, or virtual.

> Administration Server must have an internet connection to receive Kaspersky announcements.

The announcements include information of the following types:

- Security-related announcements

  Security-related announcements are intended to keep the Kaspersky applications installed in your network up-to-date and fully functional. The announcements may include information about critical updates for Kaspersky applications, fixes for found vulnerabilities, and ways to fix other issues in Kaspersky applications. By default, security-related announcements are enabled. If you do not want to receive the announcements, you can disable this feature.

  To show you the information that corresponds to your network protection configuration, Open Single Management Platform sends data to Kaspersky cloud servers and receives only those announcements that relate to the Kaspersky applications installed in your network. The data set that can be sent to the servers is described in the End User License Agreement that you accept when you install Kaspersky Security Center Administration Server.

- Marketing announcements

  Marketing announcements include information about special offers for your Kaspersky applications, advertisements, and news from Kaspersky. Marketing announcements are disabled by default. You receive this type of announcements only if you enabled Kaspersky Security Network (KSN). You can disable marketing announcements by disabling KSN.

  To show you only relevant information that might be helpful in protecting your network devices and in your everyday tasks, Open Single Management Platform sends data to Kaspersky cloud servers and receives the appropriate announcements. The data set that can be sent to the servers is described in the Processed Data section of the KSN Statement.

New information is divided into the following categories, according to importance:

1. Critical info
2. Important news
3. Warning
4. Info

When new information appears in the Kaspersky announcements section, OSMP Console displays a notification label that corresponds to the importance level of the announcements. You can click the label to view this announcement in the Kaspersky announcements section.

You can specify the Kaspersky announcements settings, including the announcement categories that you want to view and where to display the notification label. If you do not want to receive announcements, you can disable this feature.

Specifying Kaspersky announcements settings

In the Kaspersky announcements section, you can specify the Kaspersky announcements settings, including the categories of the announcements that you want to view and where to display the notification label.

To configure Kaspersky announcements:

1. In the main menu, go to **Monitoring** → **Kaspersky announcements**.
2. Click the **Settings** link.

   The Kaspersky announcement settings window opens.
3. Specify the following settings:
   - Select the importance level of the announcements that you want to view. The announcements of other categories will not be displayed.
   - Select where you want to see the notification label. The label can be displayed in all console sections, or in the **Monitoring** section and its subsections.
4. Click the **OK** button.

   The Kaspersky announcement settings are specified.

Disabling Kaspersky announcements

The Kaspersky announcements section (**Monitoring** → **Kaspersky announcements**) keeps you informed by providing information related to your version of Open Single Management Platform and managed applications installed on the managed devices. If you do not want to receive Kaspersky announcements, you can disable this feature.

The Kaspersky announcements include two types of information: security-related announcements and marketing announcements. You can disable the announcements of each type separately.

To disable security-related announcements:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Kaspersky announcements** section.

3. Switch the toggle button to the **Security-related announcements are disabled** position.

4. Click the **Save** button.

   Kaspersky announcements are disabled.

Marketing announcements are disabled by default. You receive marketing announcements only if you enabled Kaspersky Security Network (KSN). You can disable this type of announcement by disabling KSN.

To disable marketing announcements:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **KSN Proxy settings** section.

3. Disable the **Use Kaspersky Security Network Enabled** option.

4. Click the **Save** button.

   Marketing announcements are disabled.

Cloud Discovery

Open Single Management Platform allows you to monitor the use of cloud services on managed devices running Windows and to block access to cloud services that you consider unwanted. Cloud Discovery tracks user attempts to gain access to these services through both browsers and desktop applications. It also tracks user attempts to gain access to cloud services over unencrypted connections (for example, using the HTTP protocol). This feature helps you to detect and halt the use of cloud services by shadow IT.

The blocking capability is available only if you activated Open Single Management Platform under a Kaspersky Next EDR Optimum or Kaspersky Next XDR Expert license.

The blocking capability is available only if you use Kaspersky Endpoint Security 11.2 for Windows or later. Earlier versions of the security application only allow you to monitor the use of cloud services.

You can enable the Cloud Discovery feature and select the security policies or profiles for which you want to enable the feature. You can also enable or disable the feature separately in each security policy or profile. You can block access to cloud services that you do not want users to access.

To be able to block access to unwanted cloud services, make sure that the following prerequisites are met:

- You use Kaspersky Endpoint Security 11.2 for Windows or later. Earlier versions of the security application only allow you to monitor the use of cloud services.
- You have purchased a Kaspersky Next license tier that includes the ability to block access to unwanted cloud services. For details, refer to Kaspersky Next Help.

The Cloud Discovery widget and the Cloud Discovery reports display information about successful and blocked attempts to gain access to cloud services. The widget also displays the risk level of each cloud service. Open Single Management Platform gets information about the use of cloud services from all of the managed devices that are protected only by the security policies or profiles that have the feature enabled.

Enabling Cloud Discovery by using the widget

The Cloud Discovery feature allows you to get information about the use of cloud services from all of the managed devices that are protected only by the security policies that have the feature enabled. You can enable or disable Cloud Discovery for the Kaspersky Endpoint Security for Windows policy only.

There are two ways to enable the Cloud Discovery feature:

- By using the Cloud Discovery widget.
- In the properties of the Kaspersky Endpoint Security for Windows policy.

  For details on how to enable the Cloud Discovery feature in the Kaspersky Endpoint Security for Windows policy properties, refer to the Cloud Discovery section of Kaspersky Endpoint Security for Windows Help.

Note that you can disable the Cloud Discovery feature in the Kaspersky Endpoint Security for Windows policy parameters only.

> To enable Cloud Discovery, you must have the **Write** right in the **General features: Basic functionality** functional area.

To enable the Cloud Discovery feature by using the Cloud Discovery widget:

1. Go to Open Single Management Platform.
2. In the main menu, go to **Monitoring** → **Dashboard**.
3. On the **Cloud Discovery** widget, click the **Enable** button.

   > If you have Kaspersky Endpoint Security for Windows version 12.4 installed, enable the Cloud Discovery feature in the Kaspersky Endpoint Security for Windows policy properties. For details, refer to the Cloud Discovery section of Kaspersky Endpoint Security for Windows Help.

   > If you have Kaspersky Endpoint Security for Windows earlier than version 12.4, update the Kaspersky Endpoint Security for Windows plug-in to version 12.5.

4. In the **Enable Cloud Discovery** window that opens, select the security policies for which you want to enable the feature, and then click the **Enable** button.

   The following policy settings will be enabled automatically: **Inject script into web traffic to interact with web pages**, **Web Session monitor**, and **Encrypted connections scan**.

The Cloud Discovery feature is enabled and the widget is added to the dashboard.

Adding the Cloud Discovery widget to the dashboard

You can add the **Cloud Discovery** widget to the dashboard to monitor the use of cloud services on managed devices.

> To add the Cloud Discovery widget to the dashboard, you must have the **Write** right in the **General features: Basic functionality** functional area.

To add the Cloud Discovery widget to the dashboard:

1. Go to Open Single Management Platform.
2. In the main menu, go to **Monitoring** → **Dashboard**.
3. Click the **Add or restore widget** button.
4. In the list of available widgets, click the chevron icon (>) next to the **Other** category.

5. Select the **Cloud Discovery** widget, and then click the **Add** button.

   If the Cloud Discovery feature is disabled, follow the instructions in the Enabling Cloud Discovery by using the widget section.

The selected widget is added at the end of the dashboard.

Viewing information about the use of cloud services

You can view the **Cloud Discovery** widget that shows information about attempts to gain access to cloud services. The widget also displays the risk level of each cloud service. Open Single Management Platform gets information about the use of cloud services from all of the managed devices that are protected only by the security profiles that have the feature enabled.

Before viewing, make sure that:

- The Cloud Discovery widget is added to the dashboard.
- The Cloud Discovery feature is enabled.
- You have the **Read** right in the **General features: Basic functionality** functional area.

To view the Cloud Discovery widget:

1. Go to Open Single Management Platform.
2. In the main menu, go to **Monitoring** → **Dashboard**.

   The **Cloud Discovery** widget is displayed on the dashboard.
3. On the left side of the **Cloud Discovery** widget, select a category of cloud services.

   The table on the right side of the widget displays up to five services from the selected category, to which users most often try to gain access. Both successful and blocked attempts are counted.
4. On the right side of the widget, select a specific service.

   The table below displays up to ten devices that most often attempt to gain access to the service. In this table, you can generate two types of reports: report on successful access attempts and report on blocked access attempts.

   In addition, in this table you can block access to the cloud service for a specific device.

The widget displays the requested information.

From the displayed widget, you can do the following:

- Proceed to the **Monitoring** → **Reports** section, to view the Cloud Discovery reports.
- Block or allow access to the selected cloud service.

The blocking capability is available only if you activated Open Single Management Platform under a Kaspersky Next EDR Optimum or Kaspersky Next XDR Expert license.

The blocking capability is available only if you use Kaspersky Endpoint Security 11.2 for Windows or later. Earlier versions of the security application only allow you to monitor the use of cloud services.

Risk level of a cloud service

For each cloud service, Cloud Discovery provides you with a risk level. The risk level helps you determine which services do not fit the security requirements of your organization. For example, you may want to take the risk level into account when deciding whether to block access to a certain service.

The risk level is an estimated index and does not say anything about the quality of a cloud service or about the service manufacturer. The risk level is simply a recommendation from Kaspersky experts.

Risk levels of cloud services are displayed in the Cloud Discovery widget and in the list of all monitored cloud services.

Blocking access to unwanted cloud services

You can block access to cloud services that you do not want users to access. You can also allow access to cloud services that were previously blocked.

Among other considerations, you may want to take the risk level into account when deciding whether to block access to a certain service.

You can block or allow access to cloud services for a security policy or profile.

There are two ways to block access to unwanted cloud services:

- By using the Cloud Discovery widget.

  In this case, you can block access to the services one by one.

- In the properties of the Kaspersky Endpoint Security for Windows policy.

  In this case, you can block access to the services one by one or block an entire category at once.

  For details on how to enable the Cloud Discovery feature in the Kaspersky Endpoint Security for Windows policy properties, refer to the Cloud Discovery section of Kaspersky Endpoint Security for Windows Help.

To block or allow access to a cloud service by using the widget:

1. Open the Cloud Discovery widget, and then select the required cloud service.
2. In the **Top 10 devices that use the service** pane, find the security policy or profile for which you want to block or allow the service.
3. On the required line, in the **Access status in policy or profile** column, do any of the following:
   - To block the service, select **Blocked** in the drop-down list.
   - To allow the service, select **Allowed** in the drop-down list.
4. Click the **Save** button.

Access to the selected service is blocked or allowed for the security policy or profile.

Exporting events to SIEM systems

This section describes how to configure export of events to the SIEM systems.

Configuring event export to SIEM systems

Open Single Management Platform allows configuring event export to SIEM systems by one of the following methods: export to any SIEM system that uses Syslog format or export of events to SIEM systems directly from the Kaspersky Security Center database. When you complete this scenario, Administration Server sends events to a SIEM system automatically.

Prerequisites

Before you start configuration export of events in the Open Single Management Platform:

- Learn more about the methods of event export.
- Make sure that you have the values of system settings.

You can perform the steps of this scenario in any order.

The process of export of events to a SIEM system consists of the following steps:

- **Configuring the SIEM system to receive events from Open Single Management Platform**

  How-to instructions: Configuring event export in a SIEM system

- **Selecting the events that you want to export to the SIEM system**

  Mark which events you want to export to the SIEM system. First, mark the general events that occur in all managed Kaspersky applications. Then, you can mark the events for specific managed Kaspersky applications.

- **Configuring export of events to the SIEM system**

  You can export events by using one of the following methods:

  - Using TCP, UDP, or TLS over TCP protocols
  - Using export of events directly from the Kaspersky Security Center database (a set of public views is provided in the Kaspersky Security Center database; you can find the description of these public views in the klakdb.chm document)

Results

After configuring export of events to a SIEM system you can view export results if you selected events which you want to export.

Before you begin

When setting up automatic export of events in the Open Single Management Platform, you must specify some of the SIEM system settings. It is recommended that you check these settings in advance in order to prepare for setting up Open Single Management Platform.

To successfully configure automatic sending of events to a SIEM system, you must know the following settings:

- **SIEM system server address**

  The IP address of the server on which the currently used SIEM system is installed. Check this value in your SIEM system settings.

- **SIEM system server port**

  Port number used to establish a connection between Open Single Management Platform and your SIEM system server. You specify this value in the Open Single Management Platform settings and in the receiver settings of your SIEM system.

- **Protocol**

  Protocol used for transferring messages from Open Single Management Platform to your SIEM system. You specify this value in the Open Single Management Platform settings and in the receiver settings of your SIEM system.

About event export

Open Single Management Platform allows you to receive information about events that occur during the operation of Administration Server and Kaspersky applications installed on managed devices. Information about events is saved in the Administration Server database.

You can use event export within centralized systems that deal with security issues on an organizational and technical level, provide security monitoring services, and consolidate information from different solutions. These are SIEM systems, which provide real-time analysis of security alerts and events generated by network hardware and applications, or Security Operation Centers (SOCs).

These systems receive data from many sources, including networks, security, servers, databases, and applications. SIEM systems also provide functionality to consolidate monitored data in order to help you avoid missing critical events. In addition, the systems perform automated analysis of correlated events and alerts in order to notify the administrators of immediate security issues. Alerting can be implemented through a dashboard or can be sent through third-party channels such as email.

The process of exporting events from Open Single Management Platform to external SIEM systems involves two parties: an event sender, Open Single Management Platform, and an event receiver, a SIEM system. To successfully export events, you must configure this in your SIEM system and in the Open Single Management Platform. It does not matter which side you configure first. You can either configure the transmission of events in the Open Single Management Platform, and then configure the receipt of events by the SIEM system, or vice versa.

Syslog format of event export

You can send events in the Syslog format to any SIEM system. Using the Syslog format, you can relay any events that occur on the Administration Server and in Kaspersky applications that are installed on managed devices. When exporting events in the Syslog format, you can select exactly which types of events will be relayed to the SIEM system.

Receipt of events by the SIEM system

The SIEM system must receive and correctly parse the events received from Open Single Management Platform. For these purposes, you must properly configure the SIEM system. The configuration depends on the specific SIEM system utilized. However, there are a number of general steps in the configuration of all SIEM systems, such as configuring the receiver and the parser.

About configuring event export in a SIEM system

The process of exporting events from Open Single Management Platform to external SIEM systems involves two parties: an event sender—Open Single Management Platform and an event receiver—SIEM system. You must configure the export of events in your SIEM system and in the Open Single Management Platform.

The settings that you specify in the SIEM system depend on the particular system that you are using. Generally, for all SIEM systems you must set up a receiver and, optionally, a message parser to parse received events.

Setting up the receiver

To receive events sent by Open Single Management Platform, you must set up the receiver in your SIEM system. In general, the following settings must be specified in the SIEM system:

- **Export protocol**

  A message transfer protocol, either UDP, TCP, or TLS, over TCP. This protocol must be the same as the protocol you specified in Open Single Management Platform.

- **Port**

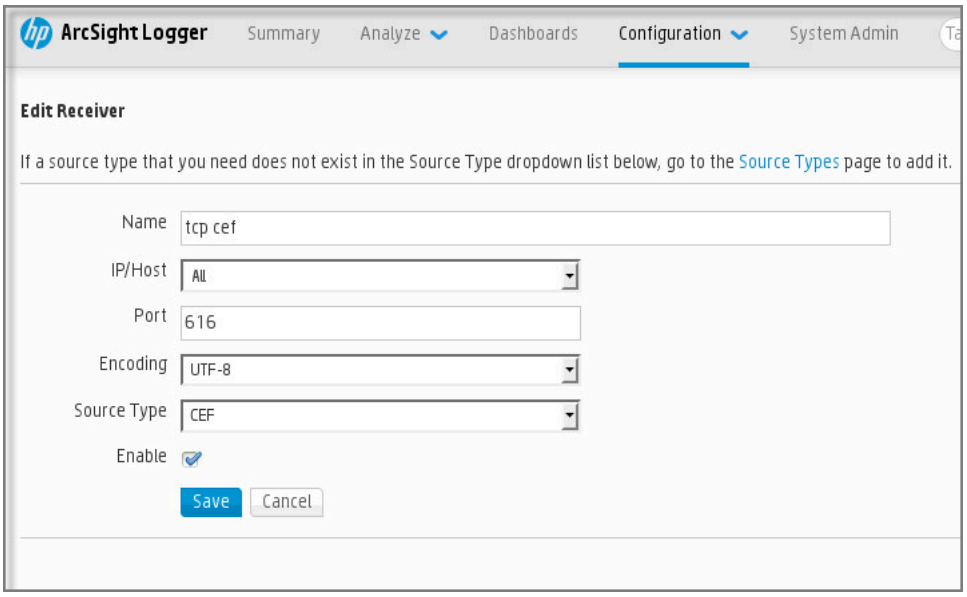
  Specify the port number to connect to Open Single Management Platform. This port must be the same as the port you specify in Open Single Management Platform during configuration with a SIEM system.

- **Data format**

  Specify the Syslog format.

Depending on the SIEM system that you use, you may have to specify some additional receiver settings.

The figure below shows the receiver setup screen in ArcSight.



Receiver setup in ArcSight

Message parser

Exported events are passed to SIEM systems as messages. These messages must be properly parsed so that information on the events can be used by the SIEM system. Message parsers are part of the SIEM system; they are used to split the contents of the message into the relevant fields, such as event ID, severity, description, parameters. This enables the SIEM system to process events received from Open Single Management Platform so that they can be stored in the SIEM system database.

Marking of events for export to SIEM systems in Syslog format

After enabling automatic export of events, you must select which events will be exported to the external SIEM system.

You can configure export of events in the Syslog format to an external system based on one of the following conditions:

- Marking general events. If you mark events to export in a policy, in the settings of an event, or in the Administration Server settings, the SIEM system will receive the marked events that occurred in all applications managed by the specific policy. If exported events were selected in the policy, you will not be able to redefine them for an individual application managed by this policy.

- Marking events for a managed application. If you mark events to export for a managed application installed on a managed device, the SIEM system will receive only the events that occurred in this application.

About marking events for export to SIEM system in the Syslog format

After enabling automatic export of events, you must select which events will be exported to the external SIEM system.

You can configure export of events in the Syslog format to an external system based on one of the following conditions:

- Marking general events. If you mark events to export in a policy, in the settings of an event, or in the Administration Server settings, the SIEM system will receive the marked events that occurred in all applications managed by the specific policy. If exported events were selected in the policy, you will not be able to redefine them for an individual application managed by this policy.

- Marking events for a managed application. If you mark events to export for a managed application installed on a managed device, the SIEM system will receive only the events that occurred in this application.

Marking events of a Kaspersky application for export in the Syslog format

If you want to export events that occurred in a specific managed application installed on the managed devices, mark the events for export in the application policy. In this case, the marked events are exported from all of the devices included in the policy scope.

To mark events for export for a specific managed application:

1. In the main menu, go to **Asset management** → **Policies**.

2. Click the policy of the application for which you want to mark events.

   The policy settings window opens.

3. Go to the **Event configuration** section.

4. Select the check boxes next to the events that you want to export to a SIEM system.

5. Click the **Mark for export to SIEM system by using Syslog** button.

   > You can also mark an event for export to a SIEM system in the **Event registration** section, which opens by clicking the link of the event.

6. A check mark (✓) appears in the **Syslog** column of the event or events that you marked for export to the SIEM system.

7. Click the **Save** button.

The marked events from the managed application are ready to be exported to a SIEM system.

You can mark which events to export to a SIEM system for a specific managed device. If previously exported events were marked in an application policy, you will not be able to redefine the marked events for a managed device.

To mark events for export for a managed device:

1. In the main menu, go to **Asset management** → **Assets (Devices)**.

   The list of managed devices is displayed.

2. Click the link with the name of the required device in the list of managed devices.

   The properties window of the selected device is displayed.

3. Go to the **Applications** section.

4. Click the link with the name of the required application in the list of applications.

5. Go to the **Event configuration** section.

6. Select the check boxes next to the events that you want to export to SIEM.

7. Click the **Mark for export to SIEM system by using Syslog** button.

   > Also, you can mark an event for export to a SIEM system in the **Event registration** section, that opens by clicking the link of the event.

8. A check mark (✓) appears in the **Syslog** column of the event or events that you marked for export to the SIEM system.

From now on, Administration Server sends the marked events to the SIEM system if export to the SIEM system is configured.

Marking general events for export in Syslog format

You can mark general events that Administration Server will export to SIEM systems by using the Syslog format.

To mark general events for export to a SIEM system:

1. In the main menu, do one of the following:

   - Click the Administration Server name.
   - Go to **Asset management** → **Policies**, and then click a link of a policy.

2. In the window that opens, go to the **Event configuration** tab.

3. Click **Mark for export to SIEM system by using Syslog**.

   > Also, you can mark an event for export to SIEM system in the **Event registration** section, that opens by clicking the link of the event.

4. A check mark (✓) appears in the **Syslog** column of the event or events that you marked for export to the SIEM system.

From now on, Administration Server sends the marked events to the SIEM system if export to the SIEM system is configured.

About exporting events using Syslog format

You can use the Syslog format to export to SIEM systems the events that occur in Administration Server and other Kaspersky applications installed on managed devices.

Syslog is a standard for message logging protocol. It permits separation of the software that generates messages, the system that stores them, and the software that reports and analyzes them. Each message is labeled with a facility code, indicating the software type that generates the message, and is assigned a severity level.

The Syslog format is defined by Request for Comments (RFC) documents published by the Internet Engineering Task Force (internet standards). The RFC 5424 standard is used to export the events from Open Single Management Platform to external systems.

In Open Single Management Platform, you can configure export of the events to the external systems using the Syslog format.

The export process consists of two steps:

1. Enabling automatic event export. At this step, Open Single Management Platform is configured so that it sends events to the SIEM system. Open Single Management Platform starts sending events immediately after you enable automatic export.

2. Selecting the events to be exported to the external system. At this step, you select which event to export to the SIEM system.

Configuring Open Single Management Platform for export of events to a SIEM system

To export events to a SIEM system, you have to configure the process of export in Open Single Management Platform.

To configure export to SIEM systems in the OSMP Console:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **SIEM** section.

3. Click the **Settings** link.

   The **Export settings** section opens.

4. Specify the settings in the **Export settings** section:

- **SIEM system server address**

  The IP address of the server on which the currently used SIEM system is installed. Check this value in your SIEM system settings.

- **SIEM system port**

  Port number used to establish a connection between Open Single Management Platform and your SIEM system server. You specify this value in the Open Single Management Platform settings and in the receiver settings of your SIEM system.

- **Protocol**

  Select the protocol to be used for transferring messages to the SIEM system. You can select TCP, UDP, or TLS over TCP protocol.

  Specify the following TLS settings if you select the TLS over TCP protocol:

  - **Server authentication**

    In the **Server authentication** field, you can select the **Trusted certificates** or **SHA fingerprints** values:

    - **Trusted certificates**. You can receive a complete certificate chain (including the root certificate) from a trusted certification authority (CA) and upload the file to Open Single Management Platform. Open Single Management Platform checks whether the certificate chain of the SIEM system server is also signed by a trusted CA or not.

      To add a trusted certificate, click the **Browse for CA certificates file** button, and then upload the certificate.

    - **SHA fingerprints**. You can specify SHA1 thumbprints of the complete certificate chain of the SIEM system (including the root certificate) in Open Single Management Platform. To add a SHA1 thumbprint, enter it in the **Thumbprints** field, and then click the **Add** button.

    By using the **Add client authentication** setting, you can generate a certificate to authenticate Open Single Management Platform. Thus, you will use a self-signed certificate issued by Open Single Management Platform. In this case, you can use both a trusted certificate and a SHA fingerprint to authenticate the SIEM system server.

  - **Add Subject name/Subject alternative name**

    Subject name is a domain name for which the certificate is received. Open Single Management Platform cannot connect to the SIEM system server if the domain name of the SIEM system server does not match the subject name of the SIEM system server certificate. However, the SIEM system server can change its domain name if the name has changed in the certificate. In this case, you can specify subject names in the **Add Subject name/Subject alternative name** field. If any of the specified subject names matches the subject name of the SIEM system certificate, Open Single Management Platform validates the SIEM system server certificate.

  - **Add client authentication**

For client authentication, you can insert your certificate or generate it in Open Single Management Platform.

- **Insert certificate**. You can use a certificate that you received from any source, for example, from any trusted CA. You must specify the certificate and its private key by using one of the following certificate types:
  - **X.509 certificate PEM**. Upload a file with a certificate in the **File with certificate** field, and a file with a private key in the **File with key** field. Both files do not depend on each other and the order of loading the files is not significant. When both files are uploaded, specify the password for decoding the private key in the **Password or certificate verification** field. The password can have an empty value if the private key is not encoded.
  - **X.509 certificate PKCS12**. Upload a single file that contains a certificate and its private key in the **File with certificate** field. When the file is uploaded, specify the password for decoding the private key in the **Password or certificate verification** field. The password can have an empty value if the private key is not encoded.
- **Generate key**. You can generate a self-signed certificate in Open Single Management Platform. As a result, Open Single Management Platform stores the generated self-signed certificate, and you can pass the public part of the certificate or SHA1-fingerprint to the SIEM system.

5. If you want, you can export archived events from the Administration Server database and set the start date from which you want to start the export of archived events:

   a. Click the **Set the export start date** link.

   b. In the section that opens, specify the start date in the **Date to start export from** field.

   c. Click the **OK** button.

6. Switch the option to the **Automatically export events to SIEM system database Enabled** position.

7. To check that the SIEM system connection is configured, click the **Check connection** button.

   The connection with the SIEM system server is established, and a test event is sent. The connection status will be displayed.

   The check is performed for TCP and TLS over TCP protocols only.

8. Click the **Save** button.

Export to a SIEM system is configured. From now on, if you configured the receiving of events in a SIEM system, Administration Server exports the marked events to a SIEM system. If you set the start date of export, Administration Server also exports the marked events stored in the Administration Server database from the specified date.

Exporting events directly from the database

You can retrieve events directly from the Open Single Management Platform database without having to use the Open Single Management Platform interface. You can either query the public views directly and retrieve the event data, or create your own views on the basis of existing public views and address them to get the data you need.

Public views

For your convenience, a set of public views is provided in the Open Single Management Platform database. You can find the description of these public views in the klakdb.chm document.

The v_akpub_ev_event public view contains a set of fields that represent the event parameters in the database. In the klakdb.chm document you can also find information on public views corresponding to other Open Single Management Platform entities, for example, devices, applications, or users. You can use this information in your queries.

This section contains instructions for executing an SQL query by means of the klsql2 utility and a query example.

To create SQL queries or database views, you can also use any other program for working with databases. Information on how to view the parameters for connecting to the Open Single Management Platform database, such as instance name and database name, is given in the corresponding section.

Executing an SQL query by using the klsql2 utility

This article describes how to use the klsql2 utility, and execute an SQL query by using this utility. Use klsql2 utility version that is included in your Open Single Management Platform version installed.

To use the klsql2 utility:

1. Go to the directory where Administration Server is installed. The default installation path is /opt/kaspersky/ksc64/sbin.

2. In this directory, create a blank file with the .sql extension.

3. Open the created .sql file in any text editor.

4. In the .sql file, type the SQL query that you want, and then save the file.

5. On the device with Administration Server installed, in the command line, type the following command to execute the SQL query from the .sql file and save the results to the result.xml file:
   ```
   sudo ./klsql2 -i src.sql -u <username> -p <password> -o result.xml
   ```
   where `<username>` and `<password>` are credentials of the user account that has access to the database.

6. If required, enter the login and password of the user account that has access to the database.

7. Open the newly created result.xml files to view the SQL query results.

You can edit the .sql file and create any SQL query to the public views. Then, from the command line, execute your query and save the results to a file.

Example of an SQL query in the klsql2 utility

This section shows an example of an SQL query, executed by means of the klsql2 utility.

The following examples illustrate retrieval of the events that occurred on devices during the last seven days, and display of the events ordered by the time they occur. The most recent events are displayed first.

Example for PostgreSQL:

```
SELECT
  /* event identifier */
  "e"."nId",

  /* time, when the event occurred */
  "e"."tmRiseTime",

  /* internal name of the event type */
  "e"."strEventType",

  /* displayed name of the event */
  "e"."wstrEventTypeDisplayName",

  /* displayed description of the event */
  "e"."wstrDescription",

  /* displayed description of the event */
  "e"."wstrGroupName",

  /* displayed name of the device, on which the event occurred */
  "h"."wstrDisplayName",
  (
    CAST(("h"."nIp" / 16777216 )& 255 ) AS VARCHAR(4))  || '.'  ||
    CAST(("h"."nIp" / 65536 )& 255 ) AS VARCHAR(4))  || '.'  ||
    CAST(("h"."nIp" / 256 )& 255 ) AS VARCHAR(4))  || '.'  ||

    /* IP address of the device, on which the event occurred */
    CAST(("h"."nIp" )& 255 ) AS VARCHAR(4))
  )  AS "strIp"
FROM "v_akpub_ev_event"  AS "e"
  INNER JOIN  "v_akpub_host"  AS "h" ON "h"."nId" = "e"."nHostId"
WHERE "e"."tmRiseTime" >= NOW() AT TIME ZONE 'utc'  + make_interval(days => CAST(-7  AS INT))
ORDER BY "e"."tmRiseTime" DESC ;
```

Example for MySQL or MariaDB:

```
SELECT

  /* event identifier */
  `e`.`nId`,

  /* time, when the event occurred */
  `e`.`tmRiseTime`,

  /* internal name of the event type */
  `e`.`strEventType`,

  /* displayed name of the event */
  `e`.`wstrEventTypeDisplayName`,

  /* displayed description of the event */
  `e`.`wstrDescription`,

  /* device group name */
  `e`.`wstrGroupName`,

  /* displayed name of the device, on which the event occurred */
  `h`.`wstrDisplayName`,
  CONCAT(
    LEFT(CAST((`h`.`nIp`  DIV  1677721) & 255) AS CHAR), 4), '.',
    LEFT(CAST((`h`.`nIp`  DIV  65536) & 255) AS CHAR), 4), '.',
    LEFT(CAST((`h`.`nIp`  DIV  256) & 255) AS CHAR), 4), '.',

    /* IP address of the device, on which the event occurred */
    LEFT(CAST((`h`.`nIp`) & 255) AS CHAR), 4)
  )  AS `strIp`
FROM `v_akpub_ev_event`  AS `e`
  INNER JOIN  `v_akpub_host`  AS `h` ON `h`.`nId` = `e`.`nHostId`
WHERE `e`.`tmRiseTime` >= ADDDATE( UTC_TIMESTAMP( ) , INTERVAL -7  DAY)
ORDER BY `e`.`tmRiseTime` DESC ;
```

Viewing the Open Single Management Platform database name

If you want to access Open Single Management Platform database by means of the MySQL, or MariaDB database management tools, you must know the name of the database in order to connect to it from your SQL script editor.

To view the name of the Open Single Management Platform database:

1. In the main menu, click the Administration Server name.

   The Administration Server properties window opens.

2. On the **General** tab, select the **Details of current database** section.

   The database name is specified in the **Database name** field. Use the database name to address the database in your SQL queries.

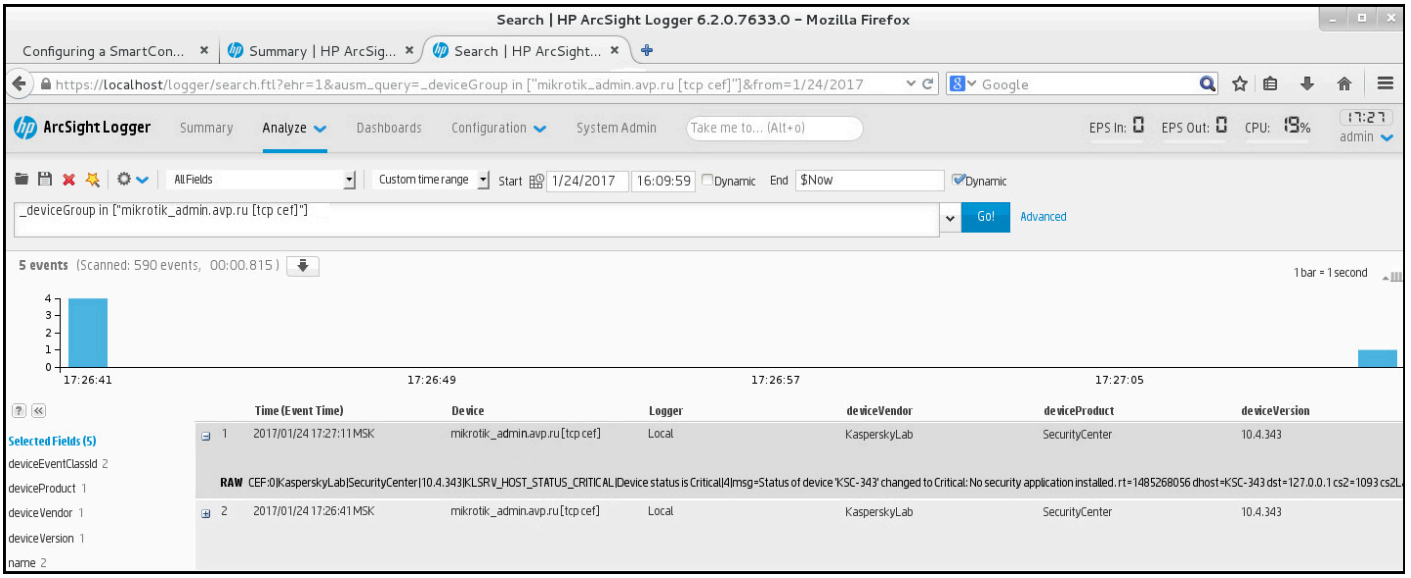
Viewing export results

You can control for successful completion of the event export procedure. To do this, check whether messages with export events are received by your SIEM system.

If the events sent from Open Single Management Platform are received and properly parsed by your SIEM system, configuration on both sides is done properly. Otherwise, check the settings you specified in Open Single Management Platform against the configuration in your SIEM system.

The figure below shows the events exported to ArcSight. For example, the first event is a critical Administration Server event: "*Device status is Critical*".

The representation of export events in the SIEM system varies according to the SIEM system you use.



Example of events

Managing object revisions

This section contains information about object revision management. Open Single Management Platform allows you to track object modification. Every time you save changes made to an object, a *revision* is created. Each revision has a number.

Application objects that support revision management include:

- Administration Server properties
- Policies
- Tasks
- Administration groups
- User accounts
- Installation packages

You can perform the following actions on object revisions:

- View a selected revision (available only for policies)
- Roll back changes made to an object to a selected revision
- Save revisions as a JSON file (available only for policies)

In the properties window of any object that supports revision management, the **Revision history** section displays a list of object revisions with the following details:

- **Revision**—Object revision number.
- **Time**—Date and time the object was modified.
- **User**—Name of the user who modified the object.
- **User device IP address**—IP address of the device from which the object was modified.
- **Web Console IP address**—IP address of OSMP Console with which the object was modified.
- **Action**—Action performed on the object.
- **Description**—Description of the revision related to the change made to the object settings.

  By default, the object revision description is blank. To add a description to a revision, select the relevant revision and click the **Edit description** button. In the opened window, enter some text for the revision description.

Viewing and saving a policy revision

Open Single Management Platform allows you to view which modifications were made to a policy over a certain period, as well as save information about these modifications in a file.

> Viewing and saving a policy revision are available if the corresponding management web plug-in supports this functionality.

To view a policy revision:

1. In the main menu, go to **Asset management** → **Policies**.
2. Click the policy for the revision that you want to view, and then go to the **Revision history** section.
3. In the list of policy revisions, click the number of the revision that you want to view.

   If the revision size is more than 10 MB, you will not be able to view it by using OSMP Console. You will be prompted to save the selected revision to a JSON file.

   If the revision size does not exceed 10 MB, a report in the HTML format with the settings of the selected policy revision is displayed. Since the report is displayed in a pop-up window, ensure that pop-ups are allowed in your browser.

To save a policy revision to a JSON file,

In the list of policy revisions, select the revision that you want to save, and then click **Save to file**.

The revision is saved to a JSON file.

Rolling back an object to a previous revision

You can roll back changes made to an object, if necessary. For example, you may have to revert the settings of a policy to their state on a specific date.

To roll back changes made to an object:

1. In the object's properties window, open the **Revision history** tab.
2. In the list of object revisions, select the revision that you want to roll back changes for.

3. Click the **Roll back** button.

4. Click **OK** to confirm the operation.

   The object is now rolled back to the selected revision. The list of object revisions displays a record of the action that was taken. The revision description displays information about the number of the revision to which you reverted the object.

Rolling back operation is available only for policy and task objects.

Deletion of objects

This section provides information about deleting objects.

You can delete objects, including the following:

- Policies
- Tasks
- Installation packages
- Virtual Administration Servers
- Users
- Security groups
- Administration groups

To delete an object:

1. Open the required section of the OSMP Console, and then select the object.

2. In the toolbar, click the **Delete** button.

3. In the window that opens, click **OK** to confirm your selection.

   The selected object is deleted.

When you delete an object, information about it remains in the database. The storage term for information about the deleted objects is the same as the storage term for object revisions (the recommended term is 90 days). You can change the storage term only if you have the **Modify** permission in the **Deleted objects** area of rights.

About deletion of client devices

When you delete a managed device from an administration group, the application moves the device to the Unassigned devices group. After device deletion, the installed Kaspersky applications—Network Agent and any security application, for example Kaspersky Endpoint Security—remain on the device.

Kaspersky Next XDR Expert handles the devices in the Unassigned devices group according to the following rules:

- If you have configured device moving rules and a device meets the criteria of a moving rule, the device is automatically moved to an administration group according to the rule.
- The device is stored in the Unassigned devices group and automatically removed from the group according to the device retention rules.

  The device retention rules do not affect the devices that have one or more drives encrypted with full disk encryption. Such devices are not deleted automatically—you can only delete them manually. If you need to delete a device with an encrypted drive, first decrypt the drive, and then delete the device.

  > When you delete a device with encrypted drive, the data required to decrypt the drive is also deleted. If you select the **I understand the risk and want to delete the selected device(s)** check box in the confirmation window that opens when you delete such devices (either from the **Unassigned devices** or the **Managed Devices** group), it means that you are aware of the subsequent data deletion.

  To decrypt the drive, the following conditions must be met:

  - The device is reconnected to Administration Server to restore the data required to decrypt the drive.
  - The device user remembers the decryption password.
  - The security application that was used to encrypt the drive, for example Kaspersky Endpoint Security for Windows, is still installed on the device.

  If the drive was encrypted by Kaspersky Disk Encryption technology, you can also try recovering data by using the FDERT Restore Utility.

> In VDI, if you have a device with encrypted drives, the information on the encrypted drive is deleted after you delete the device.

When you delete a device from the Unassigned devices group manually, the application removes the device from the list. After device deletion, the installed Kaspersky applications (if any) remain on the device. Then, if the device is still visible to Administration Server and you have configured regular network polling, Kaspersky Next XDR Expert discovers the device during the network polling and adds it back to the Unassigned devices group. Therefore, it is reasonable to delete a device manually only if the device is invisible to Administration Server.

Downloading and deleting files from Quarantine and File backups

This section gives information on how to download and how to delete files from Quarantine and File backups in OSMP Console.

Downloading files from Quarantine and File backups

You can download files from Quarantine and Backup only if one of the two conditions is met: either the **Do not disconnect from the Administration Server** option is enabled in the settings of the device, or a connection gateway is in use. Otherwise, the downloading is not possible.

To save a copy of file from Quarantine or File backups to a hard drive:

1. Do one of the following:

   - If you want to save a copy of file from Quarantine, in the main menu, go to **Operations** → **Repositories** → **Quarantine**.

   - If you want to save a copy of file from Backup, in the main menu, go to **Operations** → **Repositories** → **File backups**.

2. In the window that opens, select a file that you want to download and click **Download**.

   The download starts.

If you performed the action from the **Quarantine** section, a copy of the file that had been placed in the quarantine directory on the client device is downloaded to your computer. The file is saved in a password-protected ZIP archive. The password is the following: infected.

About removing objects from the Quarantine, File backups, or Active threats repositories

When Kaspersky security applications installed on client devices place objects to the Quarantine, Backup, or Active threats repositories, they send the information about the added objects to the **Quarantine**, **File backups**, or **Active threats** sections in Open Single Management Platform. When you open one of these sections, you can remove the required object.

For the **File backups** or **Active threats** section, select an object from the list and click the **Remove** button.

For the **Quarantine** section, you can do one of the following:

- Select the check box next to the name of an object to be removed, and then in the toolbar, click the **Delete** button.

- Click the name of the object to be removed, and in the details window that opens, in the toolbar, click the **Delete** button.

  In this case, the object is deleted, a corresponding notification is displayed on the screen, and the details window is closed.

Open Single Management Platform either removes the selected object from the list, or deletes the selected object from the repository. The action to perform is defined by the Kaspersky application that placed the selected object to the repository. The Kaspersky application is specified in the **Entry added by** field.

For details about which action is to be performed, refer to the documentation of the Kaspersky application.

Uploading files for scan

You can upload a file and then scan it, either immediately or later. Kaspersky Next XDR Expert uses the Anti-Malware Engine and YARA technologies for scanning, and also sends the file to Sandbox.

To upload a file for scan:

1. In the main menu, go to **Operations** → **Repositories** → **Files**.
2. Click **Upload**.

   The **Upload file** window opens.
3. Select a tenant to be associated with the file.
4. If you want the file to be scanned immediately after the upload is complete,make sure that the **Scan after uploading** toggle button is switched on. You can switch off this toggle button and scan the file later.
5. Upload a file by dragging and dropping or by clicking **Upload a file**. You can upload multiple files.
6. Click **Upload**.

Kaspersky Next XDR Expert starts uploading a file. After the upload is complete, Kaspersky Next XDR Expert start scanning if the **Scan after uploading** toggle button is on.

Scanning files in the File repository manually

You can manually start scanning if the file has not been scanned or if you want to rescan it. Kaspersky Next XDR Expert uses the Anti-Malware Engine and YARA technologies for scanning, and also sends the file to Sandbox.

To scan files in the File repository:

1. In the main menu, go to **Operations** → **Repositories** → **Files**.
2. Select a file or multiple files in the list.
3. Click the **Scan** button above the list.

Kaspersky Next XDR Expert start scanning the selected files. You can find the status of scanning in the **Scan status** column.

Viewing scan status and results

To view the status and results of file scanning:

1. In the main menu, go to **Operations** → **Repositories** → **Files**.

The **Scan status** column in the list of files contains basic information about scan status and results:

- **Not scanned**

  The file has not been scanned yet.

- **Scanning**

  The file is being scanned.

- **Not detected**

  The file has been scanned, no threats detected.

- **Detected**

  The file has been scanned, some threats have been detected.

2. For more detailed information, click the name of the file that you want to know more about.

   In the window that opens, you can find information about the file, including the detailed results of the file scan.

Multitenancy

Kaspersky Next XDR Expert supports a multitenancy mode. This mode enables the main administrator to provide the Kaspersky Next XDR Expert functionality to multiple clients independently, or to separate assets, application settings, and objects for different offices. Each client or office is isolated from others and is called a tenant.

Typically, the multitenancy mode is used in the following cases:

- A service provider has a number of client organizations and wants to provide the Kaspersky Next XDR Expert functionality to each client organization independently. To do this, the service provider administrator can create a tenant for each client organization.
- An administrator of a large enterprise might want to isolate assets, application settings, and objects for the offices or organization units and manage the offices or organization units independently. To do this, the administrator can create a tenant for each office or organization unit.

The multitenancy mode has the following features:

- Tenant isolation
- Cross-tenant scenarios

Tenant isolation

A tenant is isolated and managed independently from other tenants. Only users who have assigned access rights to the tenant can work within this tenant and manage it. The tenant's data, resources, and assets cannot be accessed by an administrator of another tenant unless the main administrator grants the corresponding access rights to the administrator explicitly.

For each tenant, you define a number of objects, including the following ones:

- Assets

  The asset list is unique for each tenant. Each asset can belong to one tenant only.
- Users and their access rights
- Events, alerts, and incidents
- Playbooks
- Integration with other Kaspersky applications, services, and third-party solutions

Cross-tenant scenarios

All tenants are arranged into a tenant hierarchy. By default, the tenant hierarchy contains a pre-created Root tenant at the top of the hierarchy. No other tenants can be created at the same level as the Root tenant. You create a new tenant as a child to any existing tenant, including the Root tenant. The tenant hierarchy can have any number of nesting levels.

The tenant hierarchy is used to provide cross-tenant scenarios, including the following ones:

- Inheritance and copying

  A child tenant receives the following objects from the parent tenant:

  - Users and their access rights

    Access rights are inherited down by the hierarchy and cannot be revoked on a lower level of the hierarchy.

  - Tenant settings, including integration settings

    Tenant settings are copied from a parent tenant to its child tenant. After the child tenant is created, you can reconfigure the copied settings to meet the requirements of the new tenant.

- Licensing

  A license key for Kaspersky Next XDR Expert is applied at the level of the primary Administration Server that is bound to the Root tenant. Then, the license key is automatically applied to all of the tenants in the hierarchy.

User roles

Kaspersky Next XDR Expert provides you a predefined set of user roles. You grant user rights to manage tenants by assigning user roles to the users.

User role	User right		
	Read	Write	Delete
Main administrator	✓	✓	✓
Tenant administrator	✓	✓	✓
SOC administrator	✓	✓	—
Tier 1 analyst	✓	—	—
Tier 2 analyst	✓	—	—
Junior analyst	✓	—	—
SOC manager	✓	—	—
Approver	✓	—	—
Auditor	✓	—	—
Interaction with child incidents	✓	—	—

The SLA management role allows the user to edit SLA metrics (**Time to triage**, **Time spent to respond**, **Time to resolve**) within the incident details page.

Tenants and Kaspersky Security Center Administration Servers

You can bind only physical Kaspersky Security Center Administration Servers to tenants. A link between a tenant and an Administration Server allows you to combine features of both solutions—Kaspersky Next XDR Expert and Open Single Management Platform.

Tenant filter in the application interface

In the Kaspersky Next XDR Expert interface, you can configure object lists to display only those objects that relate to the tenants that you select. The tenant filter applies to the following objects:

- Alerts in the **Alerts** section
- Incidents in the **Incidents** section
- Events in the **Threat hunting** section
- Playbooks in the **Playbooks** section

When you apply the tenant filter, the new settings are applied to all of the object types across the interface of Console that you use (OSMP Console or KUMA Console).

About binding Administration Servers to tenants

You can bind Kaspersky Security Center Administration Servers to tenants. A link between a tenant and an Administration Server allows you to relate the assets managed by the Administration Server to the tenant.

> You cannot bind virtual Administration Servers to tenants, only physical ones.

Tenants can have subtenants; therefore they are arranged into a tenant hierarchy. Administration Servers can have secondary Administration Servers; therefore they are arranged into a Server hierarchy. You cannot bind an arbitrary Server to an arbitrary tenant because this may lead to an illegal binding. For example, a user may not have access rights to a tenant in the tenant hierarchy, but the same user may have access rights to the devices of this tenant. This might happen if this user has access rights to the Administration Server 2 which is primary to the Administration Server 1 bound to the tenant. Therefore, by default, this user has inherited access rights to the Administration Server 1 and its managed devices. To eliminate such a situation, tenants and Administration Servers can only be bound to each other according to the binding rules.

There are two types of bindings:

- Explicit binding

  This binding type is established when you select an Administration Server that you want to bind to a tenant.

- Inherited binding

  When you establish explicit binding for an Administration Server that has secondary Administration Servers, the secondary Administration Servers are bound to the tenant through the inherited binding type. Therefore several Administration Servers may be bound to a tenant.

Binding rules:

- The root Administration Server is always bound to the Root tenant, you cannot remove this binding.
- A tenant may not have a bound Administration Server. Such a tenant can have subtenants, and Administration Servers can be bound to these subtenants.

- You can bind two Administration Servers which are arranged into a hierarchy only to two tenants which are arranged into a hierarchy too, and only if the hierarchy of Administration Servers matches the hierarchy of tenants.
- An Administration Server may be bound only to one tenant, explicitly or through the inherited binding type.
- When you bind an Administration Server to a tenant explicitly:
  - If the Administration Server was bound to another tenant explicitly, this binding is automatically removed.
  - If the Administration Server has secondary Administration Servers, the secondary Administration Servers are bound to the new tenant through the inherited binding type excluding those Administration Servers that were bound to their tenants explicitly. Before this operation, Kaspersky Next XDR Expert checks whether or not all of the new bindings are legal. If they are not, the binding cannot be established.
- When you remove an explicit binding between a tenant and an Administration Server (unbind Administration Server), the Administration Server and all of its secondary Administration Servers (if any) are automatically bound through the inherited binding type to the tenant to which the primary Administration Server of the selected Administration Server is bound. If some of the secondary Administration Servers are bound to their tenants explicitly, those Administration Servers keep their bindings.
- When you add a new Administration Server to the hierarchy, the Administration Server is automatically bound through the inherited binding type to the tenant to which the Server's primary Administration Server is bound.
- When you remove an Administration Server from the hierarchy and the Administration Server has an explicit binding to a tenant, this binding is removed.

Configuring integration with Open Single Management Platform

You can bind Kaspersky Security Center Administration Servers to tenants. A link between an Administration Server and a tenant allows you to relate the assets managed by the Administration Server to the tenant.

> You cannot bind virtual Administration Servers to tenants, only physical ones.

Before you begin:

- Make sure that you are familiar with the binding rules.
- You have created the tenant to which you want to bind an Administration Server.
- If required, you have added the secondary Administration Server that you want to bind to the tenant.

> To bind an Administration Server to a tenant or unbind it from the tenant, you must have a role that grants the **Write** access right to the Tenants and Integrations functional areas.

Binding an Administration Server to a tenant

To bind an Administration Server to a tenant:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant properties window opens.

3. On the **Settings** tab, in the **Kaspersky integrations** section, select **KSC**.

4. Click the **Bind Administration Server** button, and then in the window that opens, select an Administration Server that you want to bind to the tenant.

   If you want to add a new Server to the hierarchy or delete an existing one, you can do it in the Administration Server properties.

5. Click the **Bind** button.

The binding process may take a while. You can track this process in the **Binding status** column of the Administration Server list in the tenant properties window.

Unbinding an Administration Server from a tenant

To unbind an Administration Server from a tenant:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant properties window opens.

3. On the **Settings** tab, select the check box next to the Administration Server that you want to unbind from the tenant, and then click the **Unbind** button.

Viewing and editing tenants

You can use tenants to provide the Kaspersky Next XDR Expert functionality to a client organization independently, or to separate assets, application settings, and objects for different offices.

To view or edit a tenant's properties:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens. If you have only **Read** access right to this tenant, the properties will be opened in read-only mode. If you have the **Write** access right, you will be able to modify the tenant's properties.

3. Modify the tenant's properties, and then click the **Save** button.

   You can click the **Save and Close** button to save changes and close the tenant's properties window.

The tenant's properties are modified and saved.

General

The **General** tab contains general information about the tenant. You can modify the tenant's name and description.

Settings

The **Settings** tab contains the following sections:

- **Kaspersky integrations**

  This section allows you to configure integration settings for the Kaspersky applications that you want to integrate into Kaspersky Next XDR Expert for the current tenant.

- **Third-party integrations**

  This section allows you to configure integration settings for third-party applications that you want to integrate into Kaspersky Next XDR Expert for the current tenant.

- **Detection and response**

  This section allows you to configure settings and objects related to threat detection and response:

  - Segmentation rules
  - Aggregation rules
  - Mail server connection
  - Assets audit
  - Credentials management
  - Email templates
  - Scanning settings
  - Assets settings
    - Retention period for archived assets
    - Custom asset fields
    - **KSC assets import**

      All assets that are protected by this program are registered in Kaspersky Security Center. Information about assets protected by Kaspersky Security Center can be imported into KUMA. To do so, you need to configure integration between the applications in advance.

KUMA imports the following device information received from Kaspersky Security Center Network Agents:

- Basic information about the asset: name, address, time of connection to Kaspersky Security Center, hardware information, protection status, anti-virus database versions.
- Information about asset attributes: vulnerabilities; software, including the operating system; owners of the asset.

To import information about the Kaspersky Security Center assets:

1. In the main menu, go to **Asset management** → **Assets (Devices)** → **Analysis and response**.
2. Click **Import**, and then select one of the following actions:
   - To import information about the basic attributes of assets, click **Import KSC assets**.

     This opens the **Import KSC assets** window.
   - To import information about other attributes of assets, click **Import KSC assets attributes**.

     This opens the **Import KSC assets attributes** window.
3. In the drop-down list, select the tenant for which you want to perform the import.

   In this case, the program downloads information about all assets of all Kaspersky Security Center servers that have been configured to connect to the selected tenant.

   If you want to import information about all assets of all Kaspersky Security Center servers for all tenants, select **All**.
4. Click the **Import** button.

The asset information will be imported.

Also, in a tenant properties window, you can schedule regular import of asset information or run import manually.

- KICS/KATA assets import

- KPSN reputation database
- Retention period

  Retention periods for alerts and incidents depend on the Kaspersky Next XDR Expert license that you use.
- Incident management

You do not need to configure the settings of the shared tenant.

Roles

The **Roles** tab lists the users who have access rights to the tenant. You can change this list and assign user roles to the users.

Adding new tenants

Before you begin, read general information about tenants.

> To add child tenants, you must have the **Read** and **Write** rights in the **Tenants** functional area on the parent tenant or on a tenant of a higher level in the tenant hierarchy.

To add a new tenant:

1. In the main menu, go to **Settings** → **Tenants**.
2. Select the check box next to the parent tenant. The new tenant will be created as a child to the selected tenant.
3. Click the **Add** button.
4. In the **Add tenant** window that opens, enter the name of the new tenant.
5. If necessary, add a description for the new tenant.
6. Click the **Add** button.

   The new tenant appears in the tenant list.

A child tenant inherits the following objects from the parent tenant:

- Users and their access rights
- Integration settings

After a tenant is created, you can reconfigure the inherited objects to meet the requirements of the new tenant.

Assigning roles to users and groups in a tenant

You can assign XDR roles to the Kaspersky Next XDR Expert users and domain groups to provide them with sets of access rights in a tenant.

To do this, you must have one of the following XDR roles in the tenant in which you want to assign roles to users: Main administrator, SOC Administrator, or Tenant Administrator.

Since tenants are isolated and managed independently from other tenants, only users who have assigned access rights to the tenant can work within this tenant and manage it.

Access rights are inherited down in the hierarchy and cannot be revoked on a lower level of the hierarchy.

New internal user accounts in the OSMP Console become available for addition and role assignment in the tenant within five minutes after the account is created.

The rights of a user are the sum of the user's own rights and the rights of the domain groups to which the user belongs, if any. This is true when the user is included in a group directly or indirectly through the membership in a nested group. For example, if a user is a member of Group A, and this group is nested within Group B, then the user's rights is a combination of the rights assigned to this user, and the rights assigned to Group A and Group B. This means that if you assign rights to Group B, the nested groups and included users inherit these rights. However, in the case of **circular group nesting, the members of the circle as well as their child groups do not inherit the rights assigned to their parent groups**.

For example, if Group A is a member of Group B, Group B is a member of Group C, and if Group C is a member of Group A, the nesting is circular. Group A cannot inherit the rights from Groups B and C, and if Group A has a child group A1, this group does not inherit the rights from Group A. But if Group A, being a member of the circular nesting, is also included in Group Z directly, then Group A inherits the rights from Group Z (and the child group A1 does not).

To assign roles to a user or a domain group in a tenant:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Access rights** tab, and then select **Users** or **Groups**, depending on to whom you want to assign roles.

   The table of users or domain groups is displayed.

4. If necessary, configure the table:

   - Click the filter icon ( ), and then specify and apply filter criterion in the invoked menu.

     You cannot apply a filter to the **Group roles** and **Parent group roles** columns.

   - Click the settings icon ( ) to configure the columns display.

5. Click the **Add user** button or the **Add group** button, depending on the tab you selected at step 3.

6. In the window that opens, do the following:

   a. If at step 3 you selected the **Users** tab, in the **User** field, enter the user name or email address. If at step 3 you selected the **Groups** tab, in the **Group** field, specify the group name.

   b. Select the check boxes next to the roles that you want to assign to the user or the domain group.

      You can select several roles, if necessary.

   c. Click the **Add** button.

   The window is closed, and the user or the domain group is displayed in the list.

7. Click the **Save** button.

   The user or the domain group is added to the tenant and assigned roles. If necessary, you can edit the roles by clicking the user or the domain group name, and then performing the actions described at steps 6–7.

Deleting tenants

You can delete only one tenant at a time. However, if the selected tenant has child tenants, they will be deleted as well. The playbooks related to the tenant will be deleted, and information about alerts and incidents related to the tenant will become unavailable.

When you delete a tenant in OSMP Console, the following changes occur in KUMA Console:

- Tenant and its resources are deleted. Assets of the deleted tenant are archived and removed when the TTL term of the archived asset expires.
- Storage partitions related to the tenant are deleted.
- Raw events become unavailable.
- Services status of the deleted tenants change to **Gray**.

To delete a tenant, you must have the **Read** and **Write** rights in the **Tenants** functional area on the selected tenant.

You cannot delete the Root tenant and the Shared tenant that was migrated from Kaspersky Unified Monitoring and Analysis Platform.

To delete a tenant:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Select the check box next to the tenant that you want to delete. If the selected tenant has child tenants, they will be selected automatically and you cannot unselect them.

3. Click the **Delete** button.

4. Confirm the operation by typing the name of the tenant that you want to delete. If the tenant has child tenants, they are listed as tenants to be deleted as well.

The selected tenant, its child tenants (if any), and related objects are deleted.

Configuring email notifications

You can configure email notifications about events occurring in Kaspersky Next XDR Expert via Kaspersky Security Center Administration Server and an external SMTP server. To do this, you must configure connection settings to an SMTP server.

To configure connection to an SMTP server:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then in the **Detection and response** section, click **Mail server connection**.

4. In the right pane, click the **View properties** button.

   The Administration Server properties window opens with the **General** tab selected.

   > The window displays properties of the primary Administration Server and SMTP settings for the primary Administration Server, no matter to which Administration Server the tenant is bound.

5. Configure the parameters, as described at step 2 in Configuring notifications delivery.

After you configure connection to an SMTP server, the users will start receiving email messages from Kaspersky Next XDR Expert.

You can create email notifications templates that are sent to users when triggered by the specific alerts, incidents, or events. These templates can also be edited and deleted.

Creating a notification template

After you configure email notifications about events occurring in Kaspersky Next XDR Expert, you can create custom templates for these email notifications.

To create notifications templates, you must have one of the following XDR roles: Main administrator, Tenant administrator, or SOC administrator.

When you deploy Kaspersky Next XDR Expert, you have the templates for email notifications in the Root tenant. If you create a child tenant, it automatically copies the settings from the parent tenant. Since child and parent settings are not related, the changes you make in a child tenant settings do not affect the settings in the parent tenant, and vice versa.

To create an email notification template:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then in the **Detection and response** section, click **Email templates**.

   The table of the event types for which you can configure notifications templates is displayed.

4. If at step 2 you selected the Root tenant, in the **Administration Server display name** field, enter the address to be used in links to alerts and incidents in the email messages.

5. Click the **Create** button and in the drop-down list that opens, select the event type for which you need to create an email notification template.

6. In the **Create email template** panel that opens, specify the following settings:

   The available fields vary depending on the selected event type.

   - In the **Name** field, specify the name of the email notification template.
   - If you want to change the event type of the email notification template, select a new one in the **Event type** field.
   - If you want to enable an email notification for the selected event type, move the toggle button to the **Enabled** position in the **Status** field.

     You can also enable email notifications from the table of the event types. By default, email notifications are disabled.

- In the **Subject** field, specify the subject of the email notification. This field is required.

  You can also use the alert fields, incident fields, and KUMA normalized event fields in the **Subject** field, for example, `New incident in OSMP: {{ .InternalID }}, {{ .Name }}`.

- In the **Template** field, write an email notification message.

  Example of an email notification message

```
Hello,<br>
<br>
Alert {{ .InternalID }}{{ with escapeHTML .Rules.Names }} "{{ . }}"{{ else }}{{ end }} was registered
at {{ .CreatedAt }}.<br>
<br>
Details on alert:<br>
<br>
Tenant: {{ .TenantID }}<br>
Alert severity: {{ .Severity }}<br>
Alert first seen: {{ .FirstEventTime }}<br>
Alert last seen: {{ .LastEventTime }}<br>
Full information about the alert is available in OSMP web-interface: {{ link_alert . }}<br>
<br>
Open Single Management Platform
```

  You can also use alert fields, incident fields, KUMA normalized event fields, and HTML tags.

  When writing a template, you can use the following functions:

  - `date`—Defines the date and time format. The function takes the time in milliseconds (UNIX time) as the first parameter. The second parameter can be used to pass the time in the RFC standard format. The time zone cannot be changed.

```
{{range .Events }}
Service name: {{.Event.ServiceName}}
Device host name:  {{.Event.DeviceHostName}}
Message: {{.Event.Message}}
{{end}}
```

    Nested objects can have their own nested objects. You can access them using nested range functions.

  - `limit`—Limits the number of objects returned by the `range` function.

  - `link_alert`—Generates a link to the alert, with the URL specified in the **Administration Server display name** field.

  - `link_incident`—Generates a link to the incident, with the URL specified in the **Administration Server display name** field.

  - `link`—Takes the form of a link that the user can open when he/she receives the notification email.

- In the **Recipients** field, specify or select one or more email addresses for sending notifications.

- In the **Correlation rules** drop-down list, select one or more correlation rules according to which the email notification will be sent.

- In the **Notify based on severity level** field, select the minimum severity of the alert that triggers the email notification. Alerts with a severity below the specified level will not trigger an email notification.
- In the **Detection technologies** drop-down list, select one or several detection technologies that will trigger an email notification.
- If necessary, in the **Description** field, write a description of notification template.

7. Click the **Create** button.

   The **Create email template** window is closed.

8. In the tenant's properties window, save the changes.

The email notification template is created and saved. When the selected types of events occur in Kaspersky Next XDR Expert, email notifications are sent to the email addresses that you specified.

Editing and deleting a notification template

You can edit or delete an email notification template about events occurring in Kaspersky Next XDR Expert.

To edit or delete a notification template, you must have one of the following XDR roles: Main administrator, Tenant administrator, or SOC administrator.

When you deploy Kaspersky Next XDR Expert, you have the templates for email notifications in the Root tenant. If you create a child tenant, it automatically copies the settings from the parent tenant. Since child and parent settings are not related, the changes you make in a child tenant settings do not affect the settings in the parent tenant, and vice versa.

To edit an email notification template:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then in the **Detection and response** section, click **Email templates**.

4. In the **Event type** column of the table, click the event type of the notifications template that you want to edit.

5. In the **Edit email template** window that opens, change the necessary settings.

6. Save the changes.

   The **Edit email template** window is closed.

7. Save the changes.

The email notification template is edited and saved. When the selected types of events occur in Kaspersky Next XDR Expert, the template notification is sent to the email addresses that you specified.

To delete an email notification template:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then in the **Detection and response** section, click **Email templates**.

4. Select the check box next to the notification template that you want to delete.

5. Click the **Delete** button.

6. In the window that opens, confirm the deletion.

The email notification template is deleted.

Configuring the storing of credentials to access the shared network resources

The files obtained as a result of Getting disk image and Getting memory dump response actions require a large amount of free space. Thus, these files are saved to a shared network resources.

If you use several shared network resources, you have to specify credentials for each of them.

To do this, you must have one of the following XDR roles: Main administrator, SOC administrator, Tenant administrator.

To configure the credentials for accessing the shared network resources:

1. In the main menu, go to **Settings** → **Tenants**.

   The list of tenants is displayed on the screen.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Go to the **Settings** tab, and then select the **Credentials management** section.

4. Click the **Add** button.

   The **Add credentials** pane opens.

5. Specify the login and password of the account for accessing the shared network resources in the corresponding fields.

   Password cannot be deleted after you save the setting.

If necessary, you can write any comment in the **Description** field.

6. Click **Save**.

The pane is closed, the setting is saved and displayed in the table.

If necessary, you can edit the setting by clicking the link with the specified login in the table or delete the setting by selecting the check box next to the setting name in the table, and clicking **Delete**.

Configuring IOC scanning on schedule

You can configure the schedule for IOC scanning on devices with the Endpoint Agent component installed. If the scanning schedule is set, Endpoint Agents search for indicators of compromise on devices daily according to IOC rules that have scheduled scan enabled.

By default, the IOC scanning is performed daily. You can either change the start time and the duration of the scanning, or disable the daily scanning.

To configure IOC scanning on schedule:

1. In the main menu, go to **Settings** → **Tenants**.

   The tenant list opens. The list contains only the tenants to which you have at least the **Read** access right.

2. Click the name of the required tenant.

   The tenant's properties window opens.

3. Select the **Settings** tab.

4. In the **Scanning settings** section, select the **IOC scanning schedule** tab.

5. Turn on the **Daily scanning** toggle switch, and then specify the settings for the scanning schedule:

   - **Start time**. The time when the IOC scan starts. The time is specified in the time zone of the OSMP server on which you are performing the configuration.

     If an Endpoint Agent gets the new scan schedule later than the time specified in **Start time**, the next scan is initiated the next day at the time specified in **Start time**.

   - **Maximum scan duration (hours)**. Time limit for IOC scanning, in hours. When the limit is reached, Endpoint Agent finishes the scanning process regardless of its completeness.

6. Click **Save** or **Save and close** to save the settings.

The new schedule for IOC scanning becomes active immediately after changes are saved. If the scan detects a threat, the IOC alert is created.

Limitations for Kaspersky Endpoint Security for Windows

If the IOC search scope defined in an IOC rule includes the RegistryItem data type, Kaspersky Endpoint Security for Windows scans the limited set of registry keys.

Contact Technical Support

This section describes how to get technical support and the terms on which it is available.

How to get technical support

If you can't find a solution to your issue in the Kaspersky Next XDR Expert documentation or in any of the sources of information about Kaspersky Next XDR Expert, contact Kaspersky Customer Service. Technical Support specialists will answer all your questions about installing and using Kaspersky Next XDR Expert.

> Kaspersky provides support of Kaspersky Next XDR Expert during its lifecycle (see the application support lifecycle page). Before contacting Technical Support, please read the support rules.

You can contact Technical Support in one of the following ways:

- By visiting the Technical Support website
- By sending a request to Technical Support from the Kaspersky CompanyAccount portal

Technical support via Kaspersky CompanyAccount

Kaspersky CompanyAccount is a portal for companies that use Kaspersky applications. The Kaspersky CompanyAccount portal is designed to facilitate interaction between users and Kaspersky specialists through online requests. You can use Kaspersky CompanyAccount to track the status of your online requests and store a history of them as well.

You can register all of your organization's employees under a single account on Kaspersky CompanyAccount. A single account lets you centrally manage electronic requests from registered employees to Kaspersky and also manage the privileges of these employees via Kaspersky CompanyAccount.

The Kaspersky CompanyAccount portal is available in the following languages:

- English
- Spanish
- Italian
- German
- Polish
- Portuguese
- Russian

- French
- Japanese

To learn more about Kaspersky CompanyAccount, visit the Technical Support website.

Known issues

Kaspersky Next XDR Expert has a number of limitations that are not critical to the operation of the application:

- If you plan to use multiple tenants in your infrastructure, create a separate collector for each tenant.
- Kaspersky Next XDR Expert 2.0 uses an updated event data model. Playbooks based on the previous data model may not work as expected. You have to update playbooks created in Kaspersky Next XDR Expert 1.4 and earlier to ensure they work correctly with the new event data model.
- Remote terminal connections from a device with the **Encrypted connections scan** option enabled in Kaspersky Endpoint Security for Windows to another device are blocked when a self-signed certificate is used. To allow a remote terminal connection to the target device, do one of the following:
  - Use a trusted certificate on the connecting device.
  - Exclude a web address of the connecting device from encrypted connection scans.
  - Allow access to domains with untrusted certificates.
- When restoring Kaspersky Next XDR Expert from a backup by using a different database name and different certificates, the deployment stops at the `klnagchk -fullwait` validation step.
- The launch of a third-party custom script by using a playbook is supported only on a Linux device.
- When specifying a trigger for aggregation or segmentation rules, jq expressions with functions that have interdependent parameters are considered invalid. For example:

  `any(.Observables[]; .Type == "ip" and .Value == "1.1.1.1")`
- After you add or delete tenants in the **Tenants** section (**Settings** → **Tenants**), the changes to the tenant list are not synchronized with the tenant filter in the **Threat hunting** section. The tenant filter still contains the deleted tenants and does not contain the added ones.
- After you shut down the infrastructure servers of the Kubernetes cluster and then start them again, an attempt to sign in to OSMP Console returns en error.
- If you enable the **Use custom permissions** option on the **Access rights** tab in the properties of the Managed devices administration group, the client devices cannot be exported from Open Single Management Platform to KUMA.
- The playbooks that contain response actions through Kaspersky Endpoint Security for Windows are displayed as available in the playbook list even though the Kaspersky Endpoint Security for Windows web plug-in is not installed in Kaspersky Next XDR Expert.
- When you import the *Download updates to the repositories of distribution points* or *Update verification* task the **Select devices to which the task will be assigned** option is enabled. These tasks cannot be assigned to a device selection or specific devices. If you assign the *Download updates to the repositories of distribution points* or *Update verification* task to specific devices, the task will be imported incorrectly.
- In the investigation graph, rearranging nodes is performed incorrectly.

- When migrating data from the secondary Administration Server of Kaspersky Security Center Windows to the primary Administration Server of Kaspersky Next XDR Expert, the Migration wizard does not finish the **Importing data** step. This issue occurs if you create a global task on the secondary Administration Server (for example, the *Uninstall application remotely* task) and select only the **Kaspersky Security Center Administration Server** value for the **Managed applications to export** parameter in the Migration wizard.

- Receiving Kaspersky announcements is not available.

- The Administration Server properties window contains settings for mobile devices, though Kaspersky Next XDR Expert does not support management of mobile devices.

- The notifications about new versions of web plug-ins that are available to download are disabled. You can update the plug-ins by using Kaspersky Deployment Toolkit.

- After creating a new tenant, the alerts related to the tenant are sent to the server, but not displayed in the alert table. You may need to refresh the webpage to update the table data.

- In the Administration Server properties window, the **Two-factor authentication** tab displays the **Users allowed to generate secret keys** table. Since all users are allowed to generate secret keys, the table is unreasonable.

For the list of known issues of Open Single Management Platform, refer to the Kaspersky Security Center documentation.

Troubleshooting

This section provides solutions for common issues that you may encounter when using Kaspersky Next XDR Expert.

Remote terminal connections are blocked when using self-signed certificates

Remote terminal connections from a device with the **Encrypted connections scan** option enabled in Kaspersky Endpoint Security for Windows to another device are blocked when a self-signed certificate is used.

To allow a remote terminal connection to the target device, do one of the following:

- Use a trusted certificate on the connecting device.
- Exclude a web address of the connecting device from encrypted connection scans.
- Allow access to domains with untrusted certificates.

Appendices

This section provides information that complements the main document text with reference information.

Commands for components manual starting and installing

This section contains the parameters of KUMA's executable file /opt/kaspersky/kuma/kuma that can be used to manually start or install KUMA services. This may be useful for when you need to see output in the server operating system console.

Commands parameters

Commands	Description
`tools`	Start KUMA administration tools.
`collector`	Install, start, or remove a collector service.
`core`	Install, start, or uninstall the Core service.
`correlator`	Install, start, or remove a correlator service.
`agent`	Install, start, or remove an agent service.
`help`	Get information about available commands and parameters.
`license`	Get information about license.
`storage`	Start or install a Storage.
`version`	Get information about version of the program.

Flags:

`-h`, `--h` are used to get help about any kuma command. For example, `kuma <component> --help`.

Examples:

- `kuma version` is used to get version of the KUMA installer.
- `kuma core -h` – is used to get help about core command of KUMA installer.
- `kuma collector --core <address of the server where the collector should obtain its settings> --id <ID of the installed service> --api.port <port>` is used to start collector service installation.

Integrity check of KUMA files

On KUMA target hosts where KUMA services (collector, correlator, storage, event router) are installed outside a Kubernetes cluster, integrity verification is performed using scripts located in the `/opt/kaspersky/kuma/integrity/bin/ directory`. This directory contains service-specific integrity check scripts (`check_collector.sh`, `check_correlator.sh`, `check_storage.sh`, `check_event_router.sh`) as well as the script for verifying the KUMA Agent for Windows (`check_kuma_exe.sh`). It also includes verification utilities (`integrity_checker` and `integrity_checker.sh`) used internally by the integrity checking mechanism.

Integrity check scripts with logging to dmsg have been added on machines with services:

- `manual_integrity_check.sh`—allows manual integrity checks of required KUMA services (use with flags `collector`, `correlator`, `storage`, `eventRouter`).
- `systemd_integrity_check.sh`—runs integrity checks on start or restart the KUMA service.

To check the integrity of component files:

1. On the KUMA target host where a KUMA service is installed, run the following command to navigate to the directory that contains the set of scripts:

   ```
   cd /opt/kaspersky/kuma/integrity/bin
   ```

2. Run the required script (for example, `check_collector.sh`).

   The integrity of the component files is checked.

   The result of checking each component is displayed in the following format:

- The `Summary` section describes the number of scanned objects along with the scan status: integrity not confirmed / object skipped / integrity confirmed:
  - `Manifests` – the number of manifest files processed.
  - `Files` – is not used when KUMA integrity check is performed.
  - `Directories` – is not used when KUMA integrity check is performed.
  - `Registries` – is not used when KUMA integrity check is performed.
  - `Registry values` – is not used when KUMA integrity check is performed.
- Component integrity check result:
  - `SUCCEEDED` – integrity confirmed.
  - `FAILED` – integrity violated.

Example command output after a successful integrity check

```
==============================================>
Summary( failed / skipped / succeeded ):
```

```
Manifests: 0 / 0 / 1
Files: 0 / 0 / 1
Directories: 0 / 0 / 0
Registries: 0 / 0 / 0
Registry values: 0 / 0 / 0
==============================================>
SUCCEEDED
```

In the output, the number of successfully verified manifests is shown in the `Manifests` line under the succeeded column. The status `SUCCEEDED` indicates that all verified components match the signed manifests and no integrity violations were detected. If discrepancies are found, the check result is reported as `FAILED`.

The integrity of KUMA services installed outside the Kubernetes cluster is verified using this mechanism.

You can also check the integrity of installed Kaspersky Next XDR Expert components.

Normalized event data model

This section presents the KUMA normalized event data model. All events that are processed by KUMA Correlator to detect alerts must be compliant to this model.

Events that are not compliant to this data model must be imported into this format (or normalized) using Collectors.

Normalized event data model

Field name	Data type	Field size	Descriptio
The name of a field reflects its purpose. The fields can be modified.			
ApplicationProtocol	String	31 characters	Name of the application layer protocol. For example, HTTPS, SSH,
BytesIn	Number	From -9223372036854775808 to 9223372036854775807	Number of bytes received.
BytesOut	Number	From -9223372036854775808 to 9223372036854775807	Number of bytes sent.
DestinationAddress	String	45 characters	IPv4 or IPv6 address of the asset that the action will be performed xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DestinationCity	String	1023 characters	City corresponding to the IP address from the DestinationAddress
DestinationCountry	String	1023 characters	Country corresponding to the IP address from the DestinationAdd
DestinationDnsDomain	String	255 characters	The DNS portion of the fully qualified domain name of the destinat
DestinationHostName	String	1023 characters	Host name of the destination. FQDN of the destination, if available
DestinationLatitude	Float	From +/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the DestinationA
DestinationLongitude	Float	From +/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the DestinationAdd
DestinationMacAddress	String	17 characters	MAC address of the destination. For example, aa:bb:cc:dd:ee:00
DestinationNtDomain	String	255 characters	Windows Domain Name of the destination.
DestinationPort	Number	From -9223372036854775808 to 9223372036854775807	Port number of the destination.

DestinationProcessID	Number	From -9223372036854775808 to 9223372036854775807	System process ID registered on the destination.
DestinationProcessName	String	1023 characters	Name of the system process registered on the destination. For exa
DestinationRegion	String	1023 characters	Region corresponding to the IP address from the DestinationAddr
DestinationServiceName	String	1023 characters	Name of the service on the destination side. For example, sshd.
DestinationTranslatedAddress	String	45 characters	Translated IPv4 or IPv6 address of the destination. For example, 0.
DestinationTranslatedPort	Number	From -9223372036854775808 to 9223372036854775807	Port number at the destination after translation.
DestinationUserID	String	1023 characters	User ID of the destination.
DestinationUserName	String	1023 characters	User name of the destination.
DestinationUserPrivileges	String	1023 characters	Names of roles that identify user privileges at the destination. For
DeviceAction	String	63 characters	Action that was taken by the event source. For example, blocked,
DeviceAddress	String	45 characters	IPv4 or IPv6 address of the device from which the event was recei xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx
DeviceCity	String	1023 characters	City corresponding to the IP address from the DeviceAddress fiel
DeviceCountry	String	1023 characters	Country corresponding to the IP address from the DeviceAddress
DeviceDnsDomain	String	255 characters	DNS part of the fully qualified domain name of the device from wh
DeviceEventClassID	String	1023 characters	Event type ID assigned by the event source.
DeviceExternalID	String	255 characters	ID of the device or application assigned by the event source.
DeviceFacility	String	1023 characters	Value of the facility parameter set by the event source.
DeviceHostName	String	100 characters	Name of the device from which the event was received. FQDN of t
DeviceInboundinterface	String	128 characters	Name of the incoming connection interface.
DeviceLatitude	Float	From +/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the DeviceAddre
DeviceLongitude	Float	From +/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the DeviceAddress
DeviceMacAddress	String	17 characters	MAC address of the asset from which the event was received. For
DeviceNtDomain	String	255 characters	Windows Domain Name of the device.
DeviceOutboundinterface	String	128 characters	Name of the outgoing connection interface.
DevicePayloadID	String	128 characters	The payload's unique ID that is associated with the raw event.
DeviceProcessID	Number	From -9223372036854775808 to 9223372036854775807	ID of the system process on the device that generated the event.
DeviceProcessName	String	1023 characters	Name of the process.
DeviceProduct	String	63 characters	Name of the application that generated the event. The DeviceVen identify the log source.
DeviceReceiptTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the device received the event.
DeviceRegion	String	1023 characters	Region corresponding to the IP address from the DeviceAddress f
DeviceTimeZone	String	255 characters	Time zone of the device on which the event was generated.
DeviceTranslatedAddress	String	45 characters	Re-translated IPv4 or IPv6 address of the device from which the e xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

DeviceVendor	String	63 characters	Vendor name of the event source. The DeviceVendor, DeviceProd source.
DeviceVersion	String	31 characters	Product version of the event source. The DeviceVendor, DevicePr source.
EndTime	Number	From -9223372036854775808 to 9223372036854775807	Date and time (timestamp) when the event ended.
EventOutcome	String	63 characters	Result of the operation. For example, success, failure.
ExternalID	String	40 characters	Field in which the ID can be saved.
FileCreateTime	Number	From -9223372036854775808 to 9223372036854775807	File creation time.
FileHash	String	255 characters	Hash of the file. Example: CA737F1014A48F4C0B6DD43CB177B0AFD9E5169367544C49401
FileID	String	1023 characters	ID of the file.
FileModificationTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the file was last modified.
FileName	String	1023 characters	Filename without specifying the file path.
FilePath	String	1023 characters	File path, including the file name.
FilePermission	String	1023 characters	List of file permissions.
FileSize	Number	From -9223372036854775808 to 9223372036854775807	File size.
FileType	String	1023 characters	File type.
Message	String	1023 characters	Brief description of the event.
Name	String	512 characters	Name of the event.
OldFileCreateTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the OLD file was created from the event. The time is sp displayed based in the timezone of the user's browser.
OldFileHash	String	255 characters	Hash of the OLD file. Example: CA737F1014A48F4C0B6DD43CB177B0AFD9E5169367544C49401
OldFileID	String	1023 characters	ID of the OLD file.
OldFileModificationTime	Number	From -9223372036854775808 to 9223372036854775807	Time when the OLD file was last modified.
OldFileName	String	1023 characters	Name of the OLD file (without the file path).
OldFilePath	String	1023 characters	Path to the OLD file, including the file name.
OldFilePermission	String	1023 characters	List of permissions of the OLD file.
OldFileSize	Number	From -9223372036854775808 to 9223372036854775807	Size of the OLD file.
OldFileType	String	1023 characters	Type of the OLD file.
Reason	String	1023 characters	Information about the reason for the event.
RequestClientApplication	String	1023 characters	Value of the "user-agent" parameter of the http request.
RequestContext	String	2048 characters	Description of the http request context.

RequestCookies	String	1023 characters	Cookies associated with the http request.
RequestMethod	String	1023 characters	Method used when making the http request.
RequestUrl	String	1023 characters	Requested URL.
Severity	String	1023 characters	Severity. This can be the Severity field or the Level field of the raw
SourceAddress	String	45 characters	IPv4 or IPv6 address of the source. Example format: 0.0.0.0 or xxxx
SourceCity	String	1023 characters	City corresponding to the IP address from the SourceAddress fiel
SourceCountry	String	1023 characters	Country corresponding to the IP address from the SourceAddress
SourceDnsDomain	String	255 characters	The DNS portion of the fully qualified domain name of the source.
SourceHostName	String	1023 characters	Windows Domain Name of the event source device.
SourceLatitude	Float	From +/- 1.7E-308 to 1.7E+308	Longitude corresponding to the IP address from the SourceAddre
SourceLongitude	Float	From +/- 1.7E-308 to 1.7E+308	Latitude corresponding to the IP address from the SourceAddress
SourceMacAddress	String	17 characters	MAC address of the source. Format example: aa:bb:cc:dd:ee:00
SourceNtDomain	String	255 characters	Windows Domain Name of the source.
SourcePort	Number	From -9223372036854775808 to 9223372036854775807	Source port number.
SourceProcessID	Number	From -9223372036854775808 to 9223372036854775807	System process ID.
SourceProcessName	String	1023 characters	Name of the system process at the source. For example, sshd, teln
SourceRegion	String	1023 characters	Region corresponding to the IP address from the SourceAddress f
SourceServiceName	String	1023 characters	Name of the service on the source side. For example, sshd.
SourceTranslatedAddress	String	15 characters	Translated IPv4 or IPv6 address of the source. Example format: 0.0
SourceTranslatedPort	Number	From -9223372036854775808 to 9223372036854775807	Port number of the source after translation.
SourceUserID	String	1023 characters	User ID of the source.
SourceUserName	String	1023 characters	User name of the source.
SourceUserPrivileges	String	1023 characters	Names of roles that identify user privileges of the source. For exar
StartTime	Number	From -9223372036854775808 to 9223372036854775807	Date and time (timestamp) when the activity associated with the e
Tactic	String	128 characters	Name of the tactic from the MITRE ATT&CK matrix.
Technique	String	128 characters	Name of the technique from the MITRE ATT&CK matrix.
TransportProtocol	String	31 characters	Name of the Transport layer protocol of the OSI model (TCP, UDP,
Type	Number	From -9223372036854775808 to 9223372036854775807	Event type: 1 - basic, 2 - aggregated, 3 - correlation, 4 - audit, 5 - m
Fields the purpose of which can be defined by the user. The fields can be modified.			
DeviceCustomDate1	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and time value (timestamp). The time is sp displayed based in the timezone of the user's browser.
DeviceCustomDate1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomDate1 field.

DeviceCustomDate2	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and time value (timestamp). The time is sp displayed based in the timezone of the user's browser.
DeviceCustomDate2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomDate2 field.
DeviceCustomFloatingPoint1	Float	From +/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoir
DeviceCustomFloatingPoint2	Float	From +/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoir
DeviceCustomFloatingPoint3	Float	From +/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoir
DeviceCustomFloatingPoint4	Float	From +/- 1.7E-308 to 1.7E+308	Field for mapping floating point numbers.
DeviceCustomFloatingPoint4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomFloatingPoir
DeviceCustomIPv6Address1	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y
DeviceCustomIPv6Address1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address
DeviceCustomIPv6Address2	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y
DeviceCustomIPv6Address2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address
DeviceCustomIPv6Address3	String	45 characters	Field for mapping an IPv6 address value. Format example: y:y:y:y:y:y
DeviceCustomIPv6Address3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address
DeviceCustomIPv6Address4	String	45 characters	Field for mapping an IPv6 address value. For example, y:y:y:y:y:y:y:y
DeviceCustomIPv6Address4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomIPv6Address
DeviceCustomNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber1 fie
DeviceCustomNumber2	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber2 fie
DeviceCustomNumber3	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
DeviceCustomNumber3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomNumber3 fie
DeviceCustomString1	String	4000 characters	Field for mapping a string value.
DeviceCustomString1Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString1 field.
DeviceCustomString2	String	4000 characters	Field for mapping a string value.
DeviceCustomString2Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString2 field
DeviceCustomString3	String	4000 characters	Field for mapping a string value.
DeviceCustomString3Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString3 field
DeviceCustomString4	String	4000 characters	Field for mapping a string value.
DeviceCustomString4Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString4 field
DeviceCustomString5	String	4000 characters	Field for mapping a string value.

DeviceCustomString5Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString5 field
DeviceCustomString6	String	4000 characters	Field for mapping a string value.
DeviceCustomString6Label	String	1023 characters	Field for describing the purpose of the DeviceCustomString6 field
DeviceDirection	Number	From -9223372036854775808 to 9223372036854775807	Field for describing the direction of connection for an event. "0" - i
DeviceEventCategory	String	1023 characters	Event category assigned by the device that sent the event to SIEM
FlexDate1	Number, timestamp	From -9223372036854775808 to 9223372036854775807	Field for mapping a date and time value (timestamp). The time is sp displayed based in the timezone of the user's browser.
FlexDate1Label	String	128 characters	Field for describing the purpose of the FlexDate1Label field.
FlexNumber1	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
FlexNumber1Label	String	128 characters	Field for describing the purpose of the FlexNumber1Label field.
FlexNumber2	Number	From -9223372036854775808 to 9223372036854775807	Field for mapping an integer value.
FlexNumber2Label	String	128 characters	Field for describing the purpose of the FlexNumber2Label field.
FlexString1	String	1023 characters	Field for mapping a string value.
FlexString1Label	String	128 characters	Field for describing the purpose of the FlexString1Label field.
FlexString2	String	1023 characters	Field for mapping a string value.
FlexString2Label	String	128 characters	Field for describing the purpose of the FlexString2Label field.
Service fields. Cannot be edited.			
AffectedAssets	Nested [Affected] structure	-	Nested structure from which you can query alert-related assets a appear in alert events.
AggregationRuleID	String	-	ID of the aggregation rule.
AggregationRuleName	String	-	Name of the aggregation rule that processed the event.
BaseEventCount	Number	-	For an aggregated base event, this is the number of base events t correlation event, this is the number of base events that were proc correlation event.
BaseEvents	Nested [Event] list	-	Nested structure containing a list of base events. This field can be
Code	String	-	In a base event, this is the code of a process, function or operatior
CorrelationRuleID	String	-	ID of the correlation rule.
CorrelationRuleName	String	-	Name of the correlation rule that triggered the creation of the cor
DestinationAccountID	String	-	This field stores the user ID.
DestinationAssetID	String	-	This field stores the asset ID of the destination.
DeviceAssetID	String	-	This field stores the ID of the asset that sent the event to SIEM.
Extra	Nested [string:string] dictionary	-	During normalization of a raw event, this field can be used to place fields. This field can be filled in only for base events. The maximum
GroupedBy	String	-	List of names of the fields that were used for grouping in the corre
ID	String	-	Unique event ID of UUID type. The collector generates the ID for a correlator generates the ID of a correlation event. The ID never cha
Raw	String	-	Non-normalized text of the original 'raw' event. Maximum field size

ReplayID	String	-	ID of the retroscan that generated the event.
ServiceID	String	-	ID of the service instance: correlator, collector, storage.
ServiceName	String	-	Name of the microservice instance that the KUMA administrator a
SourceAccountID	String	-	This field stores the user ID.
SourceAssetID	String	-	This field stores the asset ID of the event source.
SpaceID	String	-	ID of the space.
TenantID	String	-	This field stores the ID of the tenant.
TI	Nested [string:string] dictionary	-	Field that contains categories in a dictionary format received from indicators from an event.
TICategories	map[String]	-	This field contains categories received from an external TI provide
Timestamp	Number	-	Timestamp of the base event created in the collector. Creation tin time is specified in UTC0. In the KUMA Console, the value is display

Nested `Affected` structure

Field	Data type	Description
`Assets`	Nested `[AffectedRecord] list`	List and number of assets associated with the alert.
`Accounts`	Nested `[AffectedRecord] list`	List and number of user accounts associated with the alert.

Nested `AffectedRecord` structure

Field	Data type	Description
`Value`	String	ID of the asset or user account.
`Count`	Number	The number of times an asset or user account appears in alert-related events.

Fields generated by KUMA

KUMA generates the following fields that cannot be modified: BranchID, BranchName, DestinationAccountName, DestinationAssetName, DeviceAssetName, SourceAccountName, SourceAssetName, TenantName.

Configuring the data model of a normalized event from KATA EDR

To investigate the information, the IDs of the event and the KATA/EDR process must go to certain fields of the normalized event. To build a process tree for events coming from KATA/EDR, you must configure the copying of data from the fields of the raw events to the fields of the normalized event in KUMA normalizers as follows:

1. For any KATA/EDR events, you must configure normalization with copying of the following fields:

   - The `EventType` field of the KATA/EDR event must be copied to the `DeviceEventCategory` field of the normalized KUMA event.

   - The `HostName` field of the KATA/EDR event must be copied to the `DeviceHostName` field of the normalized KUMA event.

2. For any event where DeviceProduct = 'KATA', normalization must be configured in accordance with the table below.

Normalization of event fields from KATA/EDR

KATA/EDR event field	Normalized event field
IOATag	DeviceCustomIPv6Address2
	IOATag
IOAImportance	DeviceCustomIPv6Address1
	IOAImportance
FilePath	FilePath
FileName	FileName
Md5	FileHash
FileSize	FileSize

3. For events listed in the table below, additional normalization with field copying must be configured in accordance with the table.

Additional normalization with copying of event fields from KATA/EDR

Event	Raw event field	Normalized event field
Process	UniqueParentPid	FlexString1
	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
AppLock	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
BlockedDocument	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName

Event	Raw event field	Normalized event field
Module	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
FileChange	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
Driver	HostName	DeviceHostName
	FileName	FileName
	ProductName	DeviceCustomString5
		ProductName
	ProductVendor	DeviceCustomString6
		ProductVendor
Connection	UniquePid	FlexString2
	HostName	DeviceHostName
	URI	RequestURL
	RemoteIP	DestinationAddress
	RemotePort	DestinationPort
PortListen	UniquePid	FlexString2
	HostName	DeviceHostName
	LocalIP	SourceAddress
	LocalPort	SourcePort
Registry	UniquePid	FlexString2
	HostName	DeviceHostName
	ValueName	DeviceCustomString5
		New Value Name
	KeyName	DeviceCustomString4
		New Key Name
	PreviousKeyName	FlexString2
		Old Key Name
	ValueData	DeviceCustomString6
		New Value Data
	PreviousValueData	FlexString1
		Old Value Data
	ValueType	FlexNumber1
		Value Type
	PreviousValueType	FlexNumber2
		Previous Value Type

Event	Raw event field	Normalized event field
SystemEventLog	UniquePid	FlexString2
	HostName	DeviceHostName
	OperationResult	EventOutcome
	EventId	DeviceCustomNumber3
		EventId
	EventRecordId	DeviceCustomNumber2
		EventRecordId
	Channel	DeviceCustomString6
		Channel
	ProviderName	SourceUserID
ThreatDetect	UniquePid	FlexString2
	HostName	DeviceHostName
	VerdictName	EventOutcome
	DetectedObjectType	OldFileType
	isSilent	FlexString1
		Is Silent
	RecordId	DeviceCustomString5
		Record ID
	DatabaseTimestamp	DeviceCustomDate2
		Database Timestamp
ThreatDetectProcessingResult	UniquePid	FlexString2
	HostName	DeviceHostName
	ThreatStatus	DeviceCustomString5
		Threat Status
PROCESS_INTERPRET_FILE_RUN	UniquePid	FlexString2
	HostName	DeviceHostName
	FileName	FileName
	InterpretedFilePath	OldFilePath
	InterpretedFileSize	OldFileSize
	InterpretedFileHash	OldFileHash
PROCESS_CONSOLE_INTERACTIVE_INPUT	UniquePid	FlexString2
	HostName	DeviceHostName
	InteractiveInputText	DeviceCustomString4
		Command Line
AMSI SCAN	UniquePid	FlexString2
	HostName	DeviceHostName
	ObjectContent	DeviceCustomString5
		Object Content

Asset data model

The structure of an asset is represented by fields that contain values. Fields can also contain nested structures.

Asset field	Value type	Description
`ID`	String	Asset ID.
`TenantName`	String	Tenant name.
`DeletedAt`	Number	Asset deletion date.
`CreatedAt`	Number	Asset creation date.
`TenantID`	String	Tenant ID.
`DirectCategories`	Nested list of strings	Asset categories.
`CategoryModels`	Nested `[Category]` `structure`	Changes asset categories.
`AffectedByIncidents`	Nested dictionary: [string:string `TRUE/FALSE`]	IDs of incidents.
`IPAddress`	Nested list of strings	Asset IP addresses.
`FQDN`	String	Asset FQDN.
`Weight`	Number	Asset importance.
`Deleted`	String with `TRUE/FALSE` values	Indicator of whether the asset has been marked for deletion from KUMA.
`UpdatedAt`	Number	Date of last update of the asset.
`MACAddress`	Nested list of strings	Asset MAC addresses.
`IPAddressInt`	Nested list of numbers	IP address in number format.
`Owner`	Nested [OwnerInfo] structure	Asset owner information.
`OS`	Nested [OS] structure	Asset operating system information.
`DisplayName`	String	Asset name.
`APISoft`	Nested [Software] structure	Software installed on the asset.
`APIVulns`	Nested [Vulnerability] structure	Asset vulnerabilities.
`KICSServerIp`	String	KICS for Networks server IP address.
`KICSConnectorID`	Number	KICS for Networks connector ID.
`KICSDeviceID`	Number	KICS for Networks asset ID.
`KICSStatus`	String	KICS for Networks asset status.
`KICSHardware`	Nested [KICSSystemInfo] structure	Asset hardware information received from KICS for Networks.
`KICSSoft`	Nested [KICSSystemInfo] structure	Asset software information received from KICS for Networks.
`KICSRisks`	Nested [KICSRisk] structure	Asset vulnerability information received from KICS for Networks.
`Sources`	Nested [Sources] structure	Basic information about the asset from various sources.
`FromKSC`	String with `TRUE/FALSE` values	Indicator that asset details have been imported from Kaspersky Security Center.
`NAgentID`	String	ID of the Kaspersky Security Center Network Agent from which the asset information was received.
`KSCServerFQDN`	String	FQDN of the Kaspersky Security Center Server.
`KSCInstanceID`	String	Kaspersky Security Center instance ID.

KSCMasterHostname	String	Kaspersky Security Center Server host name.
KSCGroupID	Number	Kaspersky Security Center group ID.
KSCGroupName	String	Kaspersky Security Center group name.
LastVisible	Number	Date when information about the asset was last received from Kaspersky Security Center.
Products	Nested dictionary: [string:nested [ProductInfo] structure]	Information about Kaspersky applications installed on the asset received from Kaspersky Security Center.
Hardware	Nested [Hardware] structure	Asset hardware information received from Kaspersky Security Center.
KSCSoft	Nested [Software] structure	Asset software information received from Kaspersky Security Center.
KSCVulns	Nested [Vulnerability] structure	Asset vulnerability information received from Kaspersky Security Center.

Nested Category structure

Field	Value type	Description
ID	String	Category ID.
TenantID	String	Tenant ID.
TenantName	String	Tenant name.
Parent	String	Parent category.
Path	Nested list of strings	Structure of categories.
Name	String	Category name.
UpdatedAt	Number	Last update of the category.
CreatedAt	Number	Category creation date.
Description	String	Category description.
Weight	Number	Category importance.
CategorizationKind	String	Asset category assignment type.
CategorizationAt	Number	Categorization date.
CategorizationInterval	String	Category assignment interval.

Nested OwnerInfo structure

Field	Value type	Description
DisplayName	String	Name of the asset owner.

Nested OS structure

Field	Value type	Description
Name	String	Name of the operating system.
BuildNumber	Number	Operating system version.

Nested Software structure

Field	Value type	Description
`DisplayName`	String	Software name.
`DisplayVersion`	String	Software version.
`Publisher`	String	Software publisher.
`InstallDate`	String	Installation date.
`HasMSIInstaller`	String `TRUE/FALSE`	Indicates whether the software has an MSI installer.

Nested Vulnerability structure

Field	Value type	Description
`KasperskyID`	String	Vulnerability ID assigned by Kaspersky.
`ProductName`	String	Software name.
`DescriptionURL`	String	URL containing the vulnerability description.
`RecommendedMajorPatch`	String	Recommended update.
`RecommendedMinorPatch`	String	Recommended update.
`SeverityStr`	String	Vulnerability severity.
`Severity`	Number	Vulnerability severity.
`CVE`	Nested list of strings	CVE vulnerability ID.
`ExploitExists`	String `TRUE/FALSE`	Indicates whether an exploit exists.
`MalwareExists`	String `TRUE/FALSE`	Indicates whether malware exists.

Nested KICSSystemInfo structure

Field	Value type	Description
`Model`	String	Device model.
`Version`	String	Device version.
`Vendor`	String	Vendor.

Nested KICSRisk structure

Field	Value type	Description
`ID`	Number	KICS for Networks risk ID.
`Name`	String	Risk name.
`Category`	String	Risk type.
`Description`	String	Risk description.
`DescriptionUrl`	String	Link to risk description.
`Severity`	Number	Risk severity.
`Cvss`	Number	CVSS score.

Nested Sources structure

Field	Value type	Description
`KSC`	Nested [SourceInfo] structure	Asset information received from Kaspersky Security Center.
`API`	Nested [SourceInfo] structure	Asset information received through the REST API.
`Manual`	Nested [SourceInfo] structure	Manually entered information about the asset.
`KICS`	Nested [SourceInfo] structure	Asset information received from KICS for Networks.

Nested Sources structure

Field	Value type	Description
`MACAddress`	Nested list of strings	Asset MAC addresses.
`IPAddressInt`	Nested list of numbers	IP address in number format.
`Owner`	Nested [OwnerInfo] structure	Asset owner information.
`OS`	Nested [OS] structure	Asset operating system information.
`DisplayName`	String	Asset name.
`IPAddress`	Nested list of strings	Asset IP addresses.
`FQDN`	String	Asset FQDN.
`Weight`	Number	Asset importance.
`Deleted`	String with `TRUE/FALSE` values	Indicator of whether the asset has been marked for deletion from KUMA.
`UpdatedAt`	Number	Date of last update of the asset.

Nested ProductInfo structure

Field	Value type	Description
`ProductVersion`	String	Software version.
`ProductName`	String	Software name.

Nested Hardware structure

Field	Value type	Description
`NetCards`	Nested [NetCard] structure	List of network cards of the asset.
`CPU`	Nested [CPU] structure	List of asset processors.
`RAM`	Nested [RAM] structure	Asset RAM list.
`Disk`	Nested [Disk] structure	List of asset drives.

Nested Netcard structure

Field	Value type	Description
`ID`	String	Network card ID.

`MACAddresses`	Nested list of strings	MAC addresses of the network card.
`Name`	String	Network card name.
`Manufacture`	String	Network card manufacture.
`DriverVersion`	String	Driver version.

Nested RAM structure

Field	Value type	Description
`Frequency`	String	RAM frequency.
`TotalBytes`	Number	Amount of RAM, in bytes.

Nested CPU structure

Field	Value type	Description
`ID`	String	CPU ID.
`Name`	String	CPU name.
`CoreCount`	String	Number of cores.
`CoreSpeed`	String	Frequency.

Nested Disk structure

Field	Value type	Description
`FreeBytes`	Number	Available disk space.
`TotalBytes`	Number	Total disk space.

User account data model

User account fields can be addressed from email templates and during event correlation.

Field	Value type	Description
`ID`	String	User account ID.
`ObjectGUID`	String	Active Directory attribute. User account ID in Active Directory.
`TenantID`	String	Tenant ID.
`TenantName`	String	Tenant name.
`UpdatedAt`	Number	Last update of user account.
`Domain`	String	Domain.
`CN`	String	Active Directory attribute. User name.
`DisplayName`	String	Active Directory attribute. Displayed user name.
`DistinguishedName`	String	Active Directory attribute. LDAP object name.
`EmployeeID`	String	Active Directory attribute. Employee ID.

`Mail`	String	Active Directory attribute. User email address.
`MailNickname`	String	Active Directory attribute. Alternate email address.
`Mobile`	String	Active Directory attribute. Mobile phone number.
`ObjectSID`	String	Active Directory attribute. Security ID.
`SAMAccountName`	String	Active Directory attribute. Login.
`TelephoneNumber`	String	Active Directory attribute. Phone number.
`UserPrincipalName`	String	Active Directory attribute. User principal name (UPN).
`Archived`	`TRUE/FALSE string`	Indicator that determines whether a user account is obsolete.
`MemberOf`	List of strings	Active Directory attribute. Active Directory groups joined by the user. This attribute can be used for an event search during correlation.
`PreliminarilyArchived`	`TRUE/FALSE string`	Indicator that determines whether a user account should be designated as obsolete.
`CreatedAt`	Number	User account creation date.
`SN`	String	Active Directory attribute. Last name of the user.
`SAMAccountType`	String	Active Directory attribute. User account type.
`Title`	String	Active Directory attribute. Job title of the user.
`Division`	String	Active Directory attribute. User's department.
`Department`	String	Active Directory attribute. User's division.
`Manager`	String	Active Directory attribute. User's supervisor.
`Location`	String	Active Directory attribute. User's location.
`Company`	String	Active Directory attribute. User's company.
`StreetAddress`	String	Active Directory attribute. Company address.
`PhysicalDeliveryOfficeName`	String	Active Directory attribute. Delivery address.
`ManagedObjects`	List of strings	Active Directory attribute. Objects under control of the user.
`UserAccountControl`	Number	Active Directory attribute. Active Directory account type.
`WhenCreated`	Number	Active Directory attribute. User account creation date.
`WhenChanged`	Number	Active Directory attribute. User account modification date.
`AccountExpires`	Number	Active Directory attribute. User account expiration date.
`BadPasswordTime`	Number	Active Directory attribute. Date of last unsuccessful login attempt.

Audit events

Audit events are created when certain security-related actions are completed in KUMA. These events are used to ensure system integrity.

Audit events are created on the KUMA Core server and sent to the storage of the Main tenant the correlators of the Main tenant. Therefore, at least one storage must be created in the Main tenant for storing audit events. If no storage exists, you need to create one. Audit events are sent to the KUMA Audit space, which is created automatically inside the storage.

To view audit events, you need to go to the **Events** section in KUMA and add `SELECT * FROM 'events' WHERE Type=4` to the query.

As a result of executing the query, audit events are displayed if the user role allows viewing audit events.

Event fields with general information

Every audit event has the event fields described in the table below.

Event field name	Field value
`Timestamp`	Event time.
`DeviceHostName`	The event source host. For audit events, the hostname is the hostname of the host where the kuma-core service is installed because the service is the source of events.
`DeviceTimeZone`	Timezone of the system time of the server hosting the KUMA Core in the format `+-hh:mm`.
`Type`	Type of the audit event. `Audit` or `Base`.
`TenantID`	Tenant ID.
`DeviceVendor`	`Kaspersky`
`DeviceProduct`	`KUMA`
`EndTime`	Event creation time.
`SpaceID`	`KUMA Audit`
`DeviceAction`	The action captured in the audit event.

The user has edited the set of fields settings to define sources

Event field name	Field value
`DeviceAction`	`settings updated`
`DeviceFacility`	`eventSourceIdentity`
`EventOutcome`	`succeeded`
`SourceUserName`	Login of the user that made the changes.
`SourceUserID`	ID of the user that made the changes.
`DeviceCustomString5`	Updated set of fields, | is used as the delimiter.

Key file added

Event field name	Field value
`DeviceAction`	`license key installed`
`EventOutcome`	`succeeded`
`Message`	License name.
`SourceUserID`	ID of the user that added the key file.
`SourceUserName`	User name of the user that added the key file.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

Reserve license key added

Event field name	Field value
`DeviceAction`	`reserve license key installed`
`DeviceAddress`	IP address of the device on which the reserve license key was added.
`EventOutcome`	`succeeded`
`Message`	License name.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserID`	ID of the user that added the reserve license key.
`SourceUserName`	User name of the user that added the reserve license key.
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Reserve license key deleted

Event field name	Field value
`DeviceAction`	`reserve license key uninstalled`

DeviceAddress	IP address of the device on which the reserve license key was deleted.
EventOutcome	succeeded
Message	License name.
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserID	ID of the user that deleted the reserve license key.
SourceUserName	User name of the user that deleted the reserve license key.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Application activated with an activation code

Event field name	Field value
DeviceAction	license activated
EventOutcome	succeeded
Message	License name.
SourceUserID	ID of the user that entered the activation code.
SourceUserName	User name of the user that entered the activation code.
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

License term extended

Event field name	Field value
DeviceAction	license reactivated
EventOutcome	succeeded
Message	License name.
SourceUserID	ID of the user that added the key file.
SourceUserName	User name of the user that added the key file.
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.

SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

Invalid activation code

Event field name	Field value
DeviceAction	activation code license activated
EventOutcome	failed
Message	Error text.
SourceUserID	ID of the user that entered the activation code.
SourceUserName	User name of the user that entered the activation code.
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

License code blocked

Event field name	Field value
DeviceAction	license blocked
EventOutcome	succeeded
Message	License name.
SourceUserID	ID of the user that entered the activation code.
SourceUserName	User name of the user that entered the activation code.
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

License key deleted

Event field name	Field value
DeviceAction	license uninstalled
EventOutcome	succeeded
Message	License name.

`SourceUserID`	ID of the user that deleted the key file.
`SourceUserName`	User name of the user that deleted the key file.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

EPS exceeded

Event field name	Field value
`DeviceAction`	`license limit for EPS exceeded`
`EventOutcome`	`succeeded`
`Message`	License name.
`SourceUserID`	ID of the user that added the key file or entered the activation code.
`SourceUserName`	User name of the user that added the key file or entered the activation code.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

License expiration and start of grace period

Event field name	Field value
`DeviceAction`	`license expired`
`EventOutcome`	`succeeded`
`Message`	License name.
`SourceUserID`	ID of the user that added the key file or entered the activation code.
`SourceUserName`	User name of the user that added the key file or entered the activation code.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

End of grace period

Event field name	Field value
`DeviceAction`	`license ended`
`EventOutcome`	`succeeded`
`Message`	License name.
`SourceUserID`	ID of the user that added the key file or entered the activation code.
`SourceUserName`	User name of the user that added the key file or entered the activation code.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.

Expired license key added

Event field name	Field value
`DeviceAction`	`license key installed`
`IP`	The IP address at which the key file has been added or activation code has been entered.
`EventOutcome`	`failed`
`Message`	Error text.
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserID`	ID of the user that added the key file or entered the activation code.
`SourceUserName`	User name of the user that added the key file or entered the activation code.
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Changed the set of spaces to differentiate access to events

Event field name	Field value
`DeviceAction`	• `space set was created` • `space set was deleted` • `space set was updated` (including granting and revoking access permissions)
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that modified the settings.
`SourceUserID`	ID of the user that modified the settings.
`DeviceCustomString5`	ID of the space set.
`DeviceCustomString5Label`	`set ID`
`DeviceCustomString3`	Name of the space set.
`DeviceCustomString3Label`	`space name`

Service monitoring thresholds changed

Event field name	Field value
`DeviceAction`	`settings updated`
`DeviceFacility`	`serviceAlertSettings`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that modified the settings.
`SourceUserID`	ID of the user that modified the settings.
`DeviceCustomString1`	JSON with threshold values recorded in the database.
`DeviceCustomString1Label`	`thresholds`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

KUMA Core settings modified

Event field name	Field value
DeviceAction	settings updated
DeviceFacility	coreProperties
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that modified the settings.
SourceUserID	ID of the user that modified the settings.
DeviceCustomString1	JSON with the values of the KUMA Core settings recorded in the database.
DeviceCustomString1Label	properties

GeoIP databases imported

Event field name	Field value
DeviceAction	new geobase imported
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that imported the databases.
SourceUserID	ID of the user that imported the databases.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was created

Event field name	Field value
DeviceAction	service created

EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that created the service.
SourceUserID	ID of the user that created the service.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was deleted

Event field name	Field value
DeviceAction	service deleted
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that deleted the service.
SourceUserID	ID of the user that deleted the service.
DeviceExternalID	Service ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DestinationAddress	Address of the device that was used to start the service. If the service has never been started before, the field will be empty.
DestinationHostName	The FQDN of the machine that was used to start the service. If the service has never been started before, the field will be empty.
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Service was reloaded

Event field name	Field value
`DeviceAction`	`service reloaded`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that restarted the service.
`SourceUserID`	ID of the user that restarted the service.
`DeviceExternalID`	Service ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Service was restarted

Event field name	Field value
`DeviceAction`	`service restarted`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that restarted the service.
`SourceUserID`	ID of the user that restarted the service.
`DeviceExternalID`	Service ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Service was started

Event field name	Field value
`DeviceAction`	`service started`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`DeviceExternalID`	Service ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DestinationAddress`	Address of the device where the service was started.
`DestinationHostName`	FQDN of the device where the service was started.
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Service was paired

Event field name	Field value
`DeviceAction`	`service paired`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`DeviceExternalID`	Service ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Service status was changed

Event field name	Field value
`DeviceAction`	`service status changed`
`DeviceExternalID`	Service ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DestinationAddress`	Address of the device where the service was started.
`DestinationHostName`	FQDN of the device where the service was started.
`DeviceCustomString1`	`green`, `yellow`, or `red`
`DeviceCustomString1Label`	`new status`
`DeviceCustomString2`	`green`, `yellow`, or `red`
`DeviceCustomString2Label`	`old status`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Partition was deleted by user

Event field name	Field value
`DeviceAction`	`partition deleted`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that deleted the partition.
`SourceUserID`	ID of the user that deleted the partition.
`Name`	Storage name | Tenant of the partition being moved or deleted | Name of the space to which the partition belongs.
`DeviceCustomDate1`	Data modification time (UTC). The date that is displayed in the storage partitions in the `date` column.
`DeviceCustomDate1Label`	`date of event`
`Message`	`deleted by user`

Partition automatically deleted after retention period is over

Event field name	Field value
DeviceAction	partition deleted
EventOutcome	succeeded
Name	Name of the space to which the partition belongs.
DeviceCustomDate1	Data modification time (UTC). The date that is displayed in the storage partitions in the 'date' column.
DeviceCustomDate1Label	date of event
DeviceEventCategory	This field contains information from the Event log.
SourceServiceName	scheduler
Message	deleted by retention period settings

Partition deleted by absolute retention period

Event field name	Field value
Name	Name of the space to which the partition belongs.
Message	deleted partition data
DeviceAction	applied retention policy by GB
SourceServiceName	scheduler
DeviceCustomDate1	Data modification time (UTC). The date that is displayed in the storage partitions in the 'date' column.
DeviceCustomDate1Label	date of partition
DeviceCustomNumber1	Partition size.
DeviceCustomNumber1Label	size
DeviceCustomNumber2	Number of events.
DeviceCustomNumber2Label	events
DeviceEventCategory	This field contains information from the Event log.
EventOutcome	succeeded

Partition deleted by relative retention period

Event field name	Field value
Name	Name of the space to which the partition belongs.
Message	deleted partition data
DeviceAction	applied retention policy by percent
SourceServiceName	scheduler

DeviceCustomDate1	Data modification time (UTC). The date that is displayed in the storage partitions in the 'date' column.
DeviceCustomDate1Label	date of partition
DeviceCustomNumber1	Partition size.
DeviceCustomNumber1Label	size
DeviceCustomNumber2	Number of events.
DeviceCustomNumber2Label	events
DeviceEventCategory	This field contains information from the Event log.
EventOutcome	succeeded

Partition moved to cold storage

Event field name	Field value
Name	Name of the space to which the partition belongs.
Message	moved partition data to cold storage
DeviceAction	The following values are available: • applied retention policy by days • applied retention policy by percent • applied retention policy by GB
SourceServiceName	scheduler
DeviceCustomDate1	Data modification time (UTC). The date that is displayed in the storage partitions in the 'date' column.
DeviceCustomDate1Label	date of partition
DeviceCustomNumber1	Partition size.
DeviceCustomNumber1Label	size
DeviceCustomNumber2	Number of events.
DeviceCustomNumber2Label	events
EventOutcome	succeeded
DeviceCustomString1	ID of the space.
DeviceCustomString1Label	nodeID

Error occurred while moving partition to cold storage

Event field name	Field value
Name	Name of the space to which the partition belongs.
Message	move partition data to cold storage failed: {"error":"Code: 243. DB::Exception: Move is not possible. Not enough space on 'cold_volume'. (NOT_ENOUGH_SPACE) (version 23.8.8.207)\n"}
DeviceAction	The following values are available: • applied retention policy by days • applied retention policy by percent • applied retention policy by GB
SourceServiceName	scheduler
DeviceCustomDate1	Data modification time (UTC). The date that is displayed in the storage partitions in the 'date' column.

DeviceCustomDate1Label	date of partition
DeviceCustomNumber1	Partition size.
DeviceCustomNumber1Label	size
DeviceCustomNumber2	Number of events.
DeviceCustomNumber2Label	events
EventOutcome	failed
DeviceCustomString1	ID of the space.
DeviceCustomString1Label	nodeID

Active list was successfully cleared or operation failed

Audit events for active lists are created only for actions performed by users. Audit events are not generated when the active lists are modified using correlation rules. **If you need to track such changes, you can do so using alerts**.

> If an active list is modified using a correlation rule of the **simple** type, in which the **Output** and **Loop** actions are defined, an active list modification alert will be created each time the rule is triggered.

The event can be assigned the `succeeded` or `failed` status.

Since the request to clear an active list is made over a remote connection, a data transfer error may occur at any moment: both before and after deletion.

This means that the active list may be cleared successfully, but the event is assigned the `failed` status, because EventOutcome returns the TCP/IP connection status of the request, but not the succeeded or failed status of the active list clearing.

Field values of audit event "Active list was successfully cleared or operation failed"

Event field name	Field value
DeviceAction	active list cleared
EventOutcome	succeeded or failed
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User login that was used to clear the active list.
SourceUserID	User ID that was used to clear the active list.
DeviceExternalID	Service ID whose active list was cleared.
ExternalID	Active list ID.
Name	Active list name.
Message	If EventOutcome = `failed`, an error message can be found here.
DeviceCustomString5	Service tenant ID. Some errors prevent adding tenant information to the event.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Active list item was successfully changed, or operation was unsuccessful

Audit events for active lists are created only for actions performed by users. Audit events are not generated when the active lists are modified using correlation rules. **If you need to track such changes, you can do so using alerts**.

> If an active list is modified using a correlation rule of the **simple** type, in which the **Output** and **Loop** actions are defined, an active list modification alert will be created each time the rule is triggered.

The event can be assigned the `succeeded` or `failed` status.

Since the request to change an active list item is made over a remote connection, a data transfer error may occur at any moment: both before and after the change.

This means that the active list item may be changed successfully, but the event is assigned the `failed` status, because EventOutcome returns the TCP/IP connection status of the request, but not the succeeded or failed status of the active list item change.

Field values of audit event "Active list item was successfully changed, or operation was unsuccessful"

Event field name	Field value
`DeviceAction`	`active list item changed`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User login used to change the active list item.
`SourceUserID`	User ID used to change the active list item.
`DeviceExternalID`	Service ID for which the active list is changed.
`ExternalID`	Active list ID.
`Name`	Active list name.
`DeviceCustomString1`	Key name.
`DeviceCustomString1Label`	`key`
`Message`	If EventOutcome = `failed`, an error message can be found here.
`DeviceCustomString5`	Service tenant ID. Some errors prevent adding tenant information to the event.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name
`DeviceCustomString6Label`	`tenant name`

Active list item was successfully deleted or operation failed

Audit events for active lists are created only for actions performed by users. Audit events are not generated when the active lists are modified using correlation rules. **If you need to track such changes, you can do so using alerts**.

> If an active list is modified using a correlation rule of the **simple** type, in which the **Output** and **Loop** actions are defined, an active list modification alert will be created each time the rule is triggered.

The event can be assigned the `succeeded` or `failed` status.

Since the request to delete an active list item is made over a remote connection, a data transfer error may occur at any moment: both before and after deletion.

This means that the active list item may be deleted successfully, but the event is assigned the `failed` status, because EventOutcome returns the TCP/IP connection status of the request, but not the succeeded or failed status of the active list item deletion.

Field values of audit event "Active list item was successfully deleted or operation was unsuccessful"

Event field name	Field value
`DeviceAction`	`active list item deleted`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User login that was used to delete the item from the active list.
`SourceUserID`	User ID that was used to delete the item from the active list.
`DeviceExternalID`	Service ID whose active list was cleared.
`ExternalID`	Active list ID.
`Name`	Active list name.
`DeviceCustomString1`	Key name.
`DeviceCustomString1Label`	`key`
`Message`	If EventOutcome = `failed`, an error message can be found here.
`DeviceCustomString5`	Service tenant ID. Some errors prevent adding tenant information to the event.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Active list was successfully imported or operation failed

Audit events for active lists are created only for actions performed by users. Audit events are not generated when the active lists are modified using correlation rules. **If you need to track such changes, you can do so using alerts**.

> If an active list is modified using a correlation rule of the **simple** type, in which the **Output** and **Loop** actions are defined, an active list modification alert will be created each time the rule is triggered.

Active list items are imported in parts via a remote connection.

Since the import is performed via a remote connection, a data transfer error can occur at any time: when the data is imported partially or completely. EventOutcome returns the connection status, not the import status.

Field values of audit event "Active list was successfully imported or operation failed"

Event field name	Field value
`DeviceAction`	`active list imported`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User login that was used to perform the import.
`SourceUserID`	User ID that was used to perform the import.
`DeviceExternalID`	Service ID for which an import was performed.
`ExternalID`	Active list ID.
`Name`	Active list name.
`Message`	If EventOutcome = `failed`, an error message can be found here.
`DeviceCustomString5`	Service tenant ID. Some errors prevent adding tenant information to the event.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name
`DeviceCustomString6Label`	`tenant name`

Active list was exported

Audit events for active lists are created only for actions performed by users. Audit events are not generated when the active lists are modified using correlation rules. **If you need to track such changes, you can do so using alerts**.

> If an active list is modified using a correlation rule of the **simple** type, in which the **Output** and **Loop** actions are defined, an active list modification alert will be created each time the rule is triggered.

Field values of audit event "Active list was exported"

Event field name	Field value
DeviceAction	active list exported
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to perform the export.
SourceUserID	User ID that was used to perform the export.
DeviceExternalID	Service ID for which an export was performed.
ExternalID	Active list ID.
Name	Active list name.
DeviceCustomString5	Service tenant ID. Some errors prevent adding tenant information to the event.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name
DeviceCustomString6Label	tenant name

Context table exported

Audit events for context tables are created only for actions performed by users.

Field values of audit event "Context table exported"

Event field name	Field value
DeviceAction	context table exported
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to perform the export.
SourceUserID	User ID that was used to perform the export.
DeviceExternalID	Service ID for which an export was performed.
ExternalID	Context table ID
Name	Name of the context table.
DeviceCustomString5	Service tenant ID. Some errors prevent adding tenant information to the event.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Context table successfully imported or operation failed

Audit events for context tables are created only for actions performed by users.

Context table items are imported in parts via a remote connection.

Since the import is performed via a remote connection, a data transfer error can occur at any time: when the data is imported partially or completely. EventOutcome returns the connection status, not the import status.

Field values of audit event "Context table successfully imported or operation failed"

Event field name	Field value
`DeviceAction`	`context table imported`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User login that was used to perform the import.
`SourceUserID`	User ID that was used to perform the import.
`DeviceExternalID`	Service ID for which an import was performed.
`ExternalID`	Context table ID
`Name`	Name of the context table.
`Message`	If EventOutcome = `failed`, an error message can be found here.
`DeviceCustomString5`	Service tenant ID. Some errors prevent adding tenant information to the event.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Context table item successfully modified or operation failed

Audit events for context tables are created only for actions performed by users.

The event can be assigned the `succeeded` or `failed` status.

Field values of audit event "Context table item successfully modified or operation failed"

Event field name	Field value
`DeviceAction`	`context table item is set`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.

`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that changed the context table item.
`SourceUserID`	ID of the user that changed the context table item.
`DeviceExternalID`	ID of the service whose context table was modified.
`ExternalID`	Context table ID
`Name`	Name of the context table.
`DeviceCustomString1`	Name of the record key.
`DeviceCustomString1Label`	`key`
`Message`	If EventOutcome = `failed`, an error message can be found here.
`DeviceCustomString5`	Service tenant ID. Some errors prevent adding tenant information to the event.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Context table item successfully deleted or operation failed

Audit events for context tables are created only for actions performed by users.

The event can be assigned the `succeeded` or `failed` status.

Field values of audit event "Context table item modified or operation failed"

Event field name	Field value
`DeviceAction`	`context table item deleted`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User name of the user that deleted the context table item.
`SourceUserID`	ID of the user that deleted the context table item.
`DeviceExternalID`	ID of the service for which the context table item was deleted.
`ExternalID`	Context table ID
`Name`	Name of the context table.
`DeviceCustomString1`	Name of the record key.
`DeviceCustomString1Label`	`key`
`Message`	If EventOutcome = `failed`, an error message can be found here.
`DeviceCustomString5`	Service tenant ID. Some errors prevent adding tenant information to the event.

DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Context table successfully cleared or operation failed

Audit events for context tables are created only for actions performed by users.

The event can be assigned the `succeeded` or `failed` status.

Since the request to clear a context table is made over a remote connection, a data transfer error may occur at any moment: both before and after deletion.

This means that the context table may be cleared successfully, but the event gets the `failed` status anyway, because EventOutcome returns the TCP/IP connection status of the request, but not the succeeded or failed status of the context table clearing.

Field values of audit event "Context table successfully cleared or operation failed"

Event field name	Field value
DeviceAction	context table cleared
EventOutcome	succeeded or failed
SourceTranslatedAddress	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that cleared the context table.
SourceUserID	ID of the user that cleared the context table.
DeviceExternalID	ID of the service whose context table was cleared.
ExternalID	Context table ID
Name	Name of the context table.
Message	If EventOutcome = `failed`, an error message can be found here.
DeviceCustomString5	Service tenant ID. Some errors prevent adding tenant information to the event.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Resource was added

Event field name	Field value
DeviceAction	resource added
EventOutcome	succeeded

`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User login that was used to add the resource.
`SourceUserID`	User ID that was used to add the resource.
`DeviceExternalID`	Resource ID.
`DeviceProcessName`	Resource name.
`DeviceFacility`	Resource type: • `activeList` • `agent` • `aggregationRule` • `collector` • `connection` • `connector` • `correlationRule` • `correlator` • `destination` • `dictionary` • `enrichmentRule` • `filter` • `normalizer` • `proxy` • `responseRule` • `storage`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Resource was deleted

Event field name	Field value
`DeviceAction`	`resource deleted`
`EventOutcome`	`succeeded`
`SourceTranslatedAddress`	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
`SourceUserName`	User login that was used to delete the resource.
`SourceUserID`	User ID that was used to delete the resource.
`DeviceExternalID`	Resource ID.
`DeviceProcessName`	Resource name.

DeviceFacility	Resource type: • activeList • agent • aggregationRule • collector • connection • connector • correlationRule • correlator • destination • dictionary • enrichmentRule • filter • normalizer • proxy • responseRule • storage
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Resource was updated

Event field name	Field value
DeviceAction	resource updated
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to update the resource.
SourceUserID	User ID that was used to update the resource.
DeviceExternalID	Resource ID.
DeviceProcessName	Resource name.
DeviceFacility	Resource type: • activeList • agent • aggregationRule • collector • connection • connector • correlationRule • correlator • destination • dictionary • enrichmentRule • filter • normalizer • proxy • responseRule • storage
DeviceCustomString5	Tenant ID.

DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Importing resources

Event field name	Field value
DeviceAction	import resources
EventOutcome	succeeded
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User name of the user that imported the resources.
SourceUserID	ID of the user that imported the resources.
DeviceCustomString5	ID of the tenant into which the resources were imported.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Name of the tenant into which the resources were imported.
DeviceCustomString6Label	tenant name
FileType	Import source: `file` or `repository`.
FileID	If resources are imported from a repository, the package ID. If resources are imported from a file, the system ID of the file.
ExternalID	If resources are imported from a repository, the version of the imported package. If resources are imported from a file, empty value.

Settings were updated

Event field name	Field value
DeviceAction	settings updated
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header `x-real-ip` or `x-forwarded-for`. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy server, the address of the proxy server is specified.
SourcePort	Port from which the user logged in. If the user logged in using a proxy server, the port of the proxy server is specified.
SourceUserName	User login that was used to update the settings.
SourceUserID	User ID that was used to update the settings.
DeviceFacility	Type of settings.
DeviceCustomString1	JSON with threshold values recorded in the database.
DeviceCustomString5	Tenant ID.

DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

The dictionary was successfully updated on the service or operation failed

Event field name	Field value
DeviceAction	dictionary update succeeded
EventOutcome	succeeded or failed
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User login that was used to create the service.
SourceUserID	User ID that was used to create the service.
DeviceExternalID	Service ID.
ExternalID	Dictionary ID.
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceCustomString1	Dictionary type: table, dictionary.
DeviceCustomString1Label	dictionary type
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name
Message	If EventOutcome = failed, an error message can be found here.

Dictionary entry added

Dictionary item added or updated via the API.

Field values of audit event "Dictionary entry added"

Event field name	Field value
DeviceAction	dictionary item added
EventOutcome	succeeded
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.

`SourceUserName`	User name of the user that added the item.
`SourceUserID`	ID of the user that added the item.
`DeviceExternalID`	Service ID.
`ExternalID`	Dictionary ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DeviceCustomString1`	Dictionary type: table, dictionary.
`DeviceCustomString1Label`	`dictionary type`
`DeviceCustomString2`	Record key.
`DeviceCustomString2Label`	`key`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`

Dictionary entry successfully deleted or the operation failed

Dictionary item deleted (for types: table, dictionary) via the API.

The event can be assigned the `succeeded` or `failed` status.

Field values of audit event "Dictionary entry successfully deleted or the operation failed"

Event field name	Field value
`DeviceAction`	`dictionary item deleted`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
`SourceUserName`	User name of the user that deleted the dictionary item.
`SourceUserID`	ID of the user that deleted the dictionary item.
`DeviceExternalID`	ID of the service whose dictionary item was deleted.
`ExternalID`	Dictionary ID.
`DeviceProcessName`	Service name.
`DeviceFacility`	Service type.
`DeviceCustomString1`	Dictionary type: table, dictionary.
`DeviceCustomString1Label`	`dictionary type`
`DeviceCustomString2`	Record key.
DeviceCustomString2Label	`key`
`Message`	If EventOutcome = `failed`, an error message can be found here.

DeviceCustomString5	Service tenant ID. Some errors prevent adding tenant information to the event.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Extended event schema field created

Event field name	Field value
DeviceAction	extended field created
EventOutcome	succeeded
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User name of the user that created the extended event schema field.
SourceUserID	ID of the user that created the extended event schema field.
DeviceCustomString5	Full name of the field (SA.sting_array).

Extended event schema field edited

Event field name	Field value
DeviceAction	extended field edited
EventOutcome	succeeded
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User name of the user that edited the extended event schema field.
SourceUserID	ID of the user that edited the extended event schema field.
DeviceCustomString5	Full name of the field (SA.sting_array).

Extended event schema field imported

Event field name	Field value
DeviceAction	extended field imported
EventOutcome	succeeded
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User name of the user that imported the extended event schema field.
SourceUserID	ID of the user that imported the extended event schema field.
DeviceCustomString5	Full name of the field (SA.sting_array).

Normalizer with extended event schema field imported

Event field name	Field value
DeviceAction	extended field imported
EventOutcome	succeeded
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User name of the user that imported the normalizer with an extended event schema field.
SourceUserID	ID of the user that imported the normalizer with an extended event schema field.
DeviceCustomString5	Full name of the field (SA.sting_array).

Extended event schema field deleted

Event field name	Field value
DeviceAction	extended field deleted
EventOutcome	succeeded
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	User name of the user that deleted the extended event schema field.
SourceUserID	ID of the user that deleted the extended event schema field.
DeviceCustomString5	Full name of the field (SA.sting_array).

Request sent to KIRA

Event field name	Field value
DeviceAction	KIRA task
EventOutcome	succeeded/failed
SourceUserName	User login used to send the request.
SourceUserID	User ID used to send the request.
DeviceCustomString1	The resulting string that was sent.
DeviceCustomString1Label	cmd
DeviceCustomString2	ID of the event from which the request was sent.
DeviceCustomString2Label	event
DeviceCustomString3	ID of the task created to send the request.
DeviceCustomString3Label	task

KICS/KATA response

Event field name	Field value
DeviceAction	KICS response
DeviceAssetID	asset fqdn
DeviceFacility	manual response or automatic response
EventOutcome	succeeded or failed
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
SourceAddress	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
SourcePort	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
SourceUserName	Login of the user who sent the request.
SourceUserID	ID of the user who sent the request.
DeviceCustomString3	Response rule name: Authorized, Not Authorized.
DeviceCustomString3Label	response rule name
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name
DeviceExternalID	Asset ID.
SourceHostName	Asset FQDN.
Name	Asset name.
DeviceCustomString1	List of IP addresses for the asset.
DeviceCustomString1Label	addresses
Message	If EventOutcome = failed, an error message is written to this field.
DeviceCustomString2	fdqn
DeviceCustomString2Label	fqdns

Kaspersky Automated Security Awareness Platform response

Event field name	Field value
DeviceAction	KASAP response
DeviceFacility	manual response
EventOutcome	succeeded or failed
Message	Description of the error, if an error occurred, otherwise the field is empty.
SourceTranslatedAddress	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.

`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
`SourceUserName`	Login of the user who sent the request.
`SourceUserID`	ID of the user who sent the request.
`DeviceCustomString1`	The manager of the user to whom the course is assigned.
`DeviceCustomString1Label`	`manager`
`DeviceCustomString3`	Information about the group where the user belonged. Not available for `failed`.
`DeviceCustomString3Label`	`manager`
`DeviceCustomString4`	Information about the group where the user was added.
`DeviceCustomString4Label`	`new kasap group`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`
`DestinationUserID`	ID of the Active Directory user account which causes the response.
`DestinationUserName`	Account name (sAMAccountName).
`DestinationNtDomain`	Domain of the Active Directory user account which causes the response.

KEDR response

Event field name	Field value
`DeviceAction`	`KEDR response`
`DeviceFacility`	`manual response` or `automatic response`
`EventOutcome`	`succeeded` or `failed`
`Message`	Description of the error, if an error occurred, otherwise the field is empty.
`SourceTranslatedAddress`	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
`SourceUserName`	Login of the user who sent the request.
`SourceUserID`	ID of the user who sent the request.
`SourceAssetID`	KUMA asset ID which causes the response. The value is not specified if the response is based on a hash or for all assets.
`DeviceExternalID`	The external ID assigned to KUMA in KEDR. If there is only one external ID, it is not filled in when started on user hosts.
`DeviceCustomString1`	List of IP/FQDN addresses of the asset for the host prevention rule based on the selected hash from the event card.
`DeviceCustomString1Label`	`user defined list of ips or hostnames`
`DeviceCustomString2`	Sensor ID parameter in KEDR (UUIDv4 | 'all' | 'custom').
`DeviceCustomString2Label`	`sensor id of asset in KATA/EDR`
`ServiceID`	ID of the service that caused the response. Filled in only in case of automatic response.

DeviceCustomString3	Task type name: enable_network_isolation, disable_network_isolation, enable_prevention, disable_prevention, run_process.
DeviceCustomString3Label	kedr response kind
DeviceCustomString5	Tenant ID.
DeviceCustomString5Label	tenant ID
DeviceCustomString6	Tenant name.
DeviceCustomString6Label	tenant name

Importing MITRE ATT&CK techniques and tactics

Event field name	Field value
DeviceAction	import MITRE
EventOutcome	succeeded
FileID	Name of the file to import.
SourceUserID	ID of the user which sent the import request.
SourceUserName	Login of the user which sent the import request.

Metrics alert registered for service or operation failed

Event field name	Field value
DeviceProcessName	Service name.
DeviceFacility	Service type.
DeviceExternalID	Service ID.
DeviceCustomString6Label	tenant name
DeviceCustomString6	Tenant name.
DeviceCustomString5Label	tenant ID
DeviceCustomString5	Tenant ID.
DeviceCustomString2Label	Possible values: • QPS threshold reached • Failed Insert QPS threshold reached • High distribution queue • Low disk space • Low disk partition space • Output Event Loss increasing • Disk buffer size increasing • Enrichment errors increasing • High enrichment queue • Connector log errors increasing
DeviceCustomString2	created (service creation time)
DeviceCustomString1Label	API port.
DeviceCustomNumber1	Duration of work in seconds.
DeviceAction	service alert
EventOutcome	succeeded or failed

Response in Active Directory

Event field name	Field value
`DeviceAction`	`ad response`
`DeviceFacility`	`manual response` or `automatic response`
`EventOutcome`	`succeeded` or `failed`
`SourceTranslatedAddress`	This field contains the value of the HTTP header x-real-ip or x-forwarded-for. If these headers are absent, the field will be empty.
`SourceAddress`	The address from which the user logged in. If the user logged in using a proxy, there will be a proxy address.
`SourcePort`	Port from which the user logged in. If the user logged in using a proxy, there will be a port on the proxy side.
`SourceUserName`	User name of the user that applied the response.
`SourceUserID`	ID of the user that applied the response.
`DeviceCustomString3`	Response rule name: CHANGE_PASSWORD, ADD_TO_GROUP, REMOVE_FROM_GROUP, BLOCK_USER.
`DeviceCustomString3Label`	`response rule name`
`DeviceCustomString5`	Tenant ID.
`DeviceCustomString5Label`	`tenant ID`
`DeviceCustomString6`	Tenant name.
`DeviceCustomString6Label`	`tenant name`
`DestinationUserName`	The Active Directory user account to which the response is invoked (sAMAccountName).
`DestinationNtDomain`	Domain of the Active Directory user account to which the response is invoked.
`DestinationUserID`	Account UUID in KUMA.
`FlexString1`	Information about the group where the user was added or deleted.
`FlexString1Label`	`group DN`

Backing up partitions, restoring from backup, importing partitions from an external KUMA instance

Event field name	Field value
`DeviceAction`	`backup/restore/import`
`EventOutcome`	`succeeded` or `failed`
`SourceUserName`	User name of the user that performed partition backup, restoration from backup, import of partitions from an external KUMA installation.
`SourceUserID`	ID of the user that performed partition backup, restoration from backup, import of partitions from an external KUMA installation.
`DeviceExternalID`	ID of the storage service.
`Name`	Name of the storage.
`FilePath`	Full path at which the backup file is saved. Full path from which partitions are to be restored from backup or imported.
`Message`	If `EventOutcome = failed`, an error message can be found here.

Natural language query sent for conversion to SQL query by using KIRA

Event field name	Field value
`DeviceAction`	`KIRA request`
`EventOutcome`	`succeeded/failed`
`SourceUserName`	User login used to send the request.
`SourceUserID`	User ID used to send the request.
`DeviceCustomString1`	The resulting string that was sent.
`DeviceCustomString1Label`	`query text`

Correlation rules

The file that can be downloaded by clicking the link describes the correlation rules that are included in the distribution kit. It provides the scenarios covered by rules, the conditions of their use, and the necessary sources of events.

The correlation rules described in this document are contained in the SOC_package file in the OSMP distribution kit; the password for the file is SOC_package1. Only one version of the SOC rule set can be used at a time: either Russian or English.

You can add imported correlation rules to correlators that your organization uses. Refer to the following topic for details: Step 3. Correlation.

To import the correlation rule package into KUMA:

1. In the KUMA section of OSMP Console, go to **Settings** → **Repository update**, and then set the **Update source** parameter to **Kaspersky update servers**.

   You can also configure the repository update.

2. Click **Run update** to save the update settings and manually start the *Repository update* task.

3. Go to **Task manager** to ensure that the *Repository update* task is completed.

4. Go to **Resources**, and then click **Import resources**.

5. In the **Resource import** window, select the tenant to assign the imported resources to.

6. In the **Import source** drop-down list, select **Repository**, select the SOC Content package, and then click **Import**.

The resources from the SOC Content package are imported to KUMA. For more information about importing, refer to Importing resources.

Download the description of correlation rules contained in the SOC_package.xlsx file.

Time format

KUMA supports processing information passed to the fields of the event data model with the timestamp type (EndTime, StartTime, DeviceCustomDate1, etc) in the following formats:

- "May 8, 2009 5:57:51 PM",
- "oct 7, 1970",
- "oct 7, '70",
- "oct. 7, 1970",
- "oct. 7, 70",
- "Mon Jan 2 15:04:05 2006",
- "Mon Jan 2 15:04:05 MST 2006",
- "Mon Jan 02 15:04:05 -0700 2006",
- "Monday, 02-Jan-06 15:04:05 MST",
- "Mon, 02 Jan 2006 15:04:05 MST",
- "Tue, 11 Jul 2017 16:28:13 +0200 (CEST)",
- "Mon, 02 Jan 2006 15:04:05 -0700",
- "Mon 30 Sep 2018 09:09:09 PM UTC",
- "Mon Aug 10 15:44:11 UTC+0100 2015",
- "Thu, 4 Jan 2018 17:53:36 +0000",
- "Fri Jul 03 2015 18:04:07 GMT+0100 (GMT Daylight Time)",
- "Sun, 3 Jan 2021 00:12:23 +0800 (GMT+08:00)",
- "September 17, 2012 10:09am",
- "September 17, 2012 at 10:09am PST-08",
- "September 17, 2012, 10:10:09",
- "October 7, 1970",
- "October 7th, 1970",
- "12 Feb 2006, 19:17",

- "12 Feb 2006 19:17",
- "14 May 2019 19:11:40.164",
- "7 oct 70",
- "7 oct 1970",
- "03 February 2013",
- "1 July 2013",
- "2013-Feb-03".

dd/Mon/yyyy format

- "06/Jan/2008:15:04:05 -0700",
- "06/Jan/2008 15:04:05 -0700".

mm/dd/yyyy format

- "3/31/2014",
- "03/31/2014",
- "08/21/71",
- "8/1/71",
- "4/8/2014 22:05",
- "04/08/2014 22:05",
- "4/8/14 22:05",
- "04/2/2014 03:00:51",
- "8/8/1965 12:00:00 AM",
- "8/8/1965 01:00:01 PM",
- "8/8/1965 01:00 PM",
- "8/8/1965 1:00 PM",
- "8/8/1965 12:00 AM",
- "4/02/2014 03:00:51",
- "03/19/2012 10:11:59",
- "03/19/2012 10:11:59.3186369".

yyyy/mm/dd format

- "2014/3/31",
- "2014/03/31",
- "2014/4/8 22:05",
- "2014/04/08 22:05",
- "2014/04/2 03:00:51",
- "2014/4/02 03:00:51",
- "2012/03/19 10:11:59",
- "2012/03/19 10:11:59.3186369".

yyyy:mm:dd format

- "2014:3:31",
- "2014:03:31",
- "2014:4:8 22:05",
- "2014:04:08 22:05",
- "2014:04:2 03:00:51",
- "2014:4:02 03:00:51",
- "2012:03:19 10:11:59",
- "2012:03:19 10:11:59.3186369".

Format containing Chinese characters

"2014年04月08日"

yyyy-mm-ddThh format

- "2006-01-02T15:04:05+0000",
- "2009-08-12T22:15:09-07:00",
- "2009-08-12T22:15:09",
- "2009-08-12T22:15:09.988",
- "2009-08-12T22:15:09Z",

- "2017-07-19T03:21:51:897+0100",
- "2019-05-29T08:41-04" without seconds, 2-character TZ.

yyyy-mm-dd hh:mm:ss format

- "2014-04-26 17:24:37.3186369",
- "2012-08-03 18:31:59.257000000",
- "2014-04-26 17:24:37.123",
- "2013-04-01 22:43",
- "2013-04-01 22:43:22",
- "2014-12-16 06:20:00 UTC",
- "2014-12-16 06:20:00 GMT",
- "2014-04-26 05:24:37 PM",
- "2014-04-26 13:13:43 +0800",
- "2014-04-26 13:13:43 +0800 +08",
- "2014-04-26 13:13:44 +09:00",
- "2012-08-03 18:31:59.257000000 +0000 UTC",
- "2015-09-30 18:48:56.35272715 +0000 UTC",
- "2015-02-18 00:12:00 +0000 GMT",
- "2015-02-18 00:12:00 +0000 UTC",
- "2015-02-08 03:02:00 +0300 MSK m=+0.000000001",
- "2015-02-08 03:02:00.001 +0300 MSK m=+0.000000001",
- "2017-07-19 03:21:51+00:00",
- "2014-04-26",
- "2014-04",
- "2014",
- "2014-05-11 08:20:13,787".

yyyy-mm-dd-07:00 format

"2020-07-20+08:00"

mm.dd.yyyy format

- "3.31.2014",
- "03.31.2014",
- "08.21.71".

yyyy.mm.dd format

"2014.03.30"

yyyymmdd format and similar

- "20140601",
- "20140722105203".

yymmdd hh:mm:yy format

"171113 14:14:20"

Unix timestamp format

- "1332151919",
- "1384216367189",
- "1384216367111222",
- "1384216367111222333".

Mapping fields of predefined normalizers

The file available via the download link contains a description of the field mapping of preset normalizers.

Download Description of field mapping of preset normalizers.ZIP

Glossary

Administrator host

A physical or virtual machine that is used to deploy and manage the Kubernetes cluster and Kaspersky Next XDR Expert by using KDT. KDT runs on the administrator host. If the administrator host is not included in the Kubernetes cluster, it will be used only for deployment. If the administrator host is included in the cluster, it will also act as a target host that is used for operation of Kaspersky Next XDR Expert components.

Agent

A KUMA service that is used to receive events on remote devices and forward them to KUMA collectors.

Alert

An event in the organization's IT infrastructure that was marked by Open Single Management Platform as unusual or suspicious, and that may pose a threat to the security of the organization's IT infrastructure.

Asset

A device or user of the infrastructure to be protected. If an alert or incident is detected on an asset, you can perform response actions for this asset.

Bootstrap

The basic execution environment that includes the Kubernetes cluster and infrastructure components for the function of Kaspersky Next XDR Expert. Bootstrap is included in the transport archive and it is automatically installed the during deployment of Kaspersky Next XDR Expert.

Collector

A KUMA service that receives messages from event sources, processes them, and then transmits them to a storage, correlator, and/or third-party services to identify alerts.

Configuration file

A file in the YAML format that contains the list of target hosts for the Kaspersky Next XDR Expert deployment and a set of installation parameters of the Kaspersky Next XDR Expert components. Configuration file is used by KDT.

Context

A set of access parameters that define the Kubernetes cluster that the user can select to interact with. The context also includes data for connecting to the cluster by using KDT.

Correlation rule

A KUMA resource used to recognize the defined sequences of processed events and perform specific actions after recognition.

Correlator

A KUMA service that analyzes normalized events.

Custom actions

KDT commands that allows you to perform additional operations specific to the Kaspersky Next XDR Expert components (except installation, update, deletion).

Distribution package

An archive that contains the transport archive with Kaspersky Next XDR Expert components and End User License Agreements for Kaspersky Next XDR Expert and KDT, as well as the archive with the KDT utility and templates of the configuration file and KUMA inventory file.

EPP application

Application that is installed on endpoint devices to provide protection and report their protection status to the Administration Server for centralized management. In Kaspersky Next XDR Expert, EPP applications are Kaspersky Endpoint Security for Windows, Kaspersky Endpoint Security for Linux, and Kaspersky Endpoint Security for Mac.

Event

Information security events registered on the monitored elements of the organization's IT infrastructure. For example, events include login attempts, interactions with a database, and sensor information broadcasts. Each separate event may seem uninformative, but when considered together they form a bigger picture of network activities to help identify security threats.

Event router

A KUMA service that allows you to receive streams of events from collectors and correlators and then distribute the events to specified destinations in accordance with the configured filters.

Forensics

A set of methods and technologies aimed at identifying, obtaining, and analysis of data that can be used as a digital evidence of cyberattacks or other illegal activities performed on the devices.

Incident

A container of alerts that normally indicates a true positive issue in the organization's IT infrastructure. An incident may contain a single or several alerts. By using incidents, analysts can investigate multiple alerts as a single issue.

Investigation graph

A visual analysis tool that shows the relationships between events, alerts, incidents, observables, and assets (devices). Also, the investigation graph displays the details for an incident: the corresponding alerts, users, assets and their common properties.

Kaspersky Deployment Toolkit

A utility used to deploy and manage a Kubernetes cluster, Kaspersky Next XDR Expert components, and management web plug-ins. KDT runs on the administrator host and connects to target hosts via SSH.

Kubernetes cluster

A set of hosts combined by means of Kubernetes into one computing resource. The Kubernetes cluster is used for the function of Kaspersky Next XDR Expert components (except for KUMA services). The Kubernetes cluster can include both target hosts and the administrator host.

KUMA inventory file

A file in the YAML format that contains the parameters for installation of the KUMA services that are not included in the Kubernetes cluster. The path to the KUMA inventory file is included in the configuration file that is used by KDT for the Kaspersky Next XDR Expert deployment.

KUMA services

The main components of KUMA that help the system to manage events. Services allow you to receive events from event sources and subsequently bring them to a common form that is convenient for finding correlation, as well as for storage and manual analysis. KUMA services are agents, collectors, correlators, storages, and event routers that are installed on the hosts that are located outside the Kubernetes cluster.

Lightweight Nagent (LWNGT)

A protocol for interaction with Kaspersky Endpoint Security on mobile devices. The LWNGT (also called Mobile protocol) functions as Network Agent without actually installing Network Agent on mobile devices.

Multitenancy

A mode that enables the main administrator to provide the Kaspersky Next XDR Expert functionality to multiple clients independently, or to separate assets, application settings, and objects for different offices. Also the multitenancy mode allows you to copy and inherit tenant settings and objects from the parent tenant and automatically apply a license key for Kaspersky Next XDR Expert to all of the tenants in the hierarchy.

Network Agent

An Open Single Management Platform component that enables interaction between the Administration Server and Kaspersky applications that are installed on a specific network node (workstation or server). This component is common to all of the company's applications for Microsoft® Windows®. Separate versions of Network Agent exist for Kaspersky applications developed for Unix-like OS and macOS.

Network Location Awareness (NLA)

A Windows service that helps an operating system identify the current network. NLA detects network changes and adjusts the security configuration of the device.

Node

A physical or virtual machine on which Kaspersky Next XDR Expert is deployed. There are primary and worker nodes. The primary node is intended for managing the cluster, storing metadata, and distributing of the workload. The worker nodes are intended for performing the workload of the Kaspersky Next XDR Expert components.

Normalized event

An event that is processed in accordance with the KUMA normalized event data model.

Observables

Objects related to the alert and incident, such as MD5 and SHA256 hashes, IP address, URL, Domain name, UserName, or HostName.

Playbook

An object that responds to alerts or incidents according to the specified algorithm (playbook algorithm). Playbooks allow you to automate workflows and reduce the time it takes to process alerts and incidents.

Playbook algorithm

An algorithm that includes a sequence of response actions that help analyze and handle alerts or incidents.

Registry

Infrastructure component that stores the application containers and is used for the installation and storing of the Kaspersky Next XDR Expert components.

Response actions

Actions that are launched within playbooks.

Segmentation rules

Rules that allow you to automatically split related alerts into different incidents based on specified conditions.

Storage

A KUMA service that is used to store normalized events so that they can be quickly and continually accessed from KUMA for the purpose of extracting analytical data.

Target hosts

Physical or virtual machines that are used to deploy Kaspersky Next XDR Expert. Target hosts are included in the Kubernetes cluster. The Kaspersky Next XDR Expert components work on these hosts.

Tenant

A logical entity that corresponds to an organization unit (a client or an office) to which the Kaspersky Next XDR Expert functionality is provided. Each tenant can include assets, users and their access rights, events, alerts, incidents, playbooks, and integration with other Kaspersky applications, services, and third-party solutions. Also a tenant defines a set of available operations on the included objects.

Threat development chain

A series of steps that trace the stages of a cyber attack. Threat development chain allows you to analyze the reasons of the threat. To create a threat development chain, the managed application transfers data from the device to Administration Server through Network Agent.

Transport archive

An archive that contains Kaspersky Next XDR Expert components, management web plug-ins, and End User License Agreements for Kaspersky Next XDR Expert and KDT. The transport archive is included in the distribution package.

Information about third-party code

Information about third-party code is contained in the files legal_notices_ksmp.txt and legal_notices_kuma.txt on the device that acts as an operator node. The files are located in the `~/kaspersky_eula` directory of the user that runs the deployment of Kaspersky Next XDR Expert.

Trademark notices

Registered trademarks and service marks are the property of their respective owners.

Adobe, Flash, PostScript are either registered trademarks or trademarks of Adobe in the United States and/or other countries.

AMD, AMD64 are trademarks or registered trademarks of Advanced Micro Devices, Inc.

Amazon, Amazon EC2, Amazon Web Services, AWS, and AWS Marketplace are trademarks of Amazon.com, Inc. or its affiliates.

Apache, and Apache Cassandra are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Apple, App Store, AppleScript, Carbon, FileVault, iPhone, Mac, Mac OS, macOS, OS X, Safari and QuickTime are trademarks of Apple Inc.

Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

The Bluetooth word, mark and logos are owned by Bluetooth SIG, Inc.

LTS, and Ubuntu are registered trademarks of Canonical Ltd.

Check Point NGFW is a trademark or registered trademark of Check Point Software Technologies Ltd. or its affiliates.

Cisco, IOS, and Snort are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Citrix, XenServer are either registered trademarks or trademarks of Cloud Software Group, Inc., and/or its subsidiaries in the United States and/or other countries.

Citrix NetScaler is either a registered trademark or a trademark of Cloud Software Group, Inc., and/or its subsidiaries in the United States and/or other countries.

Cloudflare, the Cloudflare logo, and Cloudflare Workers are trademarks and/or registered trademarks of Cloudflare, Inc. in the United States and other jurisdictions.

The Grafana Word Mark and Grafana Logo are either registered trademarks/service marks or trademarks/service marks of Coding Instinct AB, in the United States and other countries and are used with Coding Instinct's permission. We are not affiliated with, endorsed or sponsored by Coding Instinct, or the Grafana community.

CorelDRAW is a trademark or registered trademark of Corel Corporation and/or its subsidiaries in Canada, the United States and/or other countries.

Dell Technologies and other trademarks are trademarks of Dell Inc. or its subsidiaries.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

Elasticsearch is a trademark of Elasticsearch BV, registered in the U.S. and in other countries.

ESET is a trademark or registered trademark of ESET spol. s r.o. or respective ESET entity.

F5 is a trademark of F5 Networks, Inc. in the U.S. and in certain other countries.

Firebird is a registered trademark of the Firebird Foundation.

Fortinet, FortiGate, FortiMail, FortiSOAR are either registered trademarks or trademarks of Fortinet, Inc. in the United States and/or other countries.

FreeBSD is a registered trademark of The FreeBSD Foundation.

Google, Android, Chrome, Dalvik, Firebase, Google Chrome, Google Maps, Google Play, Google Public DNS are trademarks of Google LLC.

HUAWEI, EulerOS, Huawei Eudemon are trademarks of Huawei Technologies Co., Ltd.

ViPNet is a registered trademark of Infotecs.

IBM, Guardium, InfoSphere, QRadar are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

Intel, Insider are trademarks of Intel Corporation or its subsidiaries.

Node.js is a trademark of Joyent, Inc.

Juniper, Juniper Networks, and JUNOS are trademarks or registered trademarks of Juniper Networks, Inc. in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Kubernetes is a registered trademark of The Linux Foundation in the United States and other countries.

Microsoft, Access, Active Directory, ActiveSync, ActiveX, BitLocker, Excel, Halo, Hyper-V, InfoPath, Internet Explorer, Lync, Microsoft Edge, MS-DOS, MultiPoint, Office 365, OneNote, Outlook, PowerPoint, PowerShell, Segoe, SharePoint, Skype, SQL Server, Tahoma, Visio, Win32, Windows, Windows Media, Windows Mobile, Windows Phone, Windows PowerShell, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies.

CVE is a registered trademark of The MITRE Corporation.

Mozilla, Firefox are trademarks of the Mozilla Foundation in the U.S. and other countries.

NetApp is a trademark or a registered trademark of NetApp, Inc. in the United States and/or other countries.

Netskope, the Netskope logo, and other Netskope product names referenced herein are trademarks of Netskope, Inc. and/or one of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.

NetWare is a registered trademark of Novell Inc. in the United States and other countries.

Novell is a registered trademark of Novell Enterprises Inc. in the United States and other countries.

OpenSSL is a trademark owned by the OpenSSL Software Foundation.

Oracle, Java, and JavaScript are registered trademarks of Oracle and/or its affiliates.

OpenVPN is a registered trademark of OpenVPN, Inc.

Parallels, the Parallels logo, and Coherence are trademarks or registered trademarks of Parallels International GmbH.

PROOFPOINT is a trademark of Proofpoint, Inc. in the U.S. and other countries.

Chef is a trademark or registered trademark of Progress Software Corporation and/or one of its subsidiaries or affiliates in the U.S. and/or other countries.

Puppet is a trademark or registered trademark of Puppet, Inc.

Python is a trademark or registered trademark of the Python Software Foundation.

Ansible is a registered trademark of Red Hat, Inc. in the United States and other countries.

Red Hat, CentOS, Fedora, Red Hat Enterprise Linux are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

The Trademark BlackBerry is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries.

Samsung is a trademark of SAMSUNG in the United States or other countries.

Sendmail and other names and product names are trademarks or registered trademarks of Sendmail, Inc.

Debian is a registered trademark of Software in the Public Interest, Inc.

Dameware trademark is the exclusive property of SolarWinds Worldwide, LLC or its affiliates, is registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries.

Splunk is a trademark and registered trademark of Splunk Inc. in the United States and other countries.

The CommuniGate Pro name is a trademark or registered trademark of Stalker Software, Inc.

SUSE is a registered trademark of SUSE LLC in the United States and other countries.

Symantec is a trademark or registered trademark of Symantec Corporation or its affiliates in the U.S. and other countries.

Symbian trademark is owned by the Symbian Foundation Ltd.

OpenAPI is a trademark of The Linux Foundation.

Rocky Linux is a trademark of The Rocky Enterprise Software Foundation.

Trend Micro is a trademark or registered trademark of Trend Micro Incorporated.

The names, images, logos and pictures identifying UserGate's products and services are proprietary marks of UserGate and/or its subsidiaries or affiliates, and the products themselves are proprietary to UserGate.

VMware, VMware ESXi, VMware Horizon, VMware vCenter, VMware vSphere, VMware Workstation, Carbon Black are registered trademarks and/or trademarks of VMware, Inc. in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

ClickHouse is a trademark of YANDEX LLC.

Zabbix is a registered trademark of Zabbix SIA.

The MITRE Corporation Terms of Use

LICENSE

The MITRE Corporation (MITRE) hereby grants you a non-exclusive, royalty-free license to use ATT&CK® for research, development, and commercial purposes. Any copy you make for such purposes is authorized provided that you reproduce MITRE's copyright designation and this license in any such copy.

"© 2025 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation."

DISCLAIMERS

MITRE does not claim ATT&CK enumerates all possibilities for the types of actions and behaviors documented as part of its adversary model and framework of techniques. Using the information contained within ATT&CK to address or cover full categories of techniques will not guarantee full defensive coverage as there may be undisclosed techniques or variations on existing techniques not documented by ATT&CK.

ALL DOCUMENTS AND THE INFORMATION CONTAINED THEREIN ARE PROVIDED ON AN "AS IS" BASIS AND THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE MITRE CORPORATION, ITS BOARD OF TRUSTEES, OFFICERS, AGENTS, AND EMPLOYEES, DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.